# Bayesian Nonparametric Clustering based on Dirichlet Processes


Sivakumar Murugiah

Department of Statistics

University College London

A thesis submitted for the degree of

*Doctor of Philosophy*

August, 2010

# Declaration

I, Sivakumar Murugiah, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Signed:

Date:

# Acknowledgements

My greatest debt goes to my supervisor Professor Trevor Sweeting, Head of the Department of Statistical Science at University College London, whose fantastic breadth of knowledge and intuition, I have been privileged enough to have access to, guided me in countless meetings. He helped me to structure and clarify my thoughts and made some valuable contributions on my research. Professor Trevor Sweeting provided invaluable support in reading nearly all versions of my thesis. I am very much indebted to him for being thorough, where I was sloppy, and ingenious. I also thank him for numerous occasions where his encouragement helped increase my stamina to continue on this long journey. I am also very thankful to Professor Tom Fearn, Director of Research, at University College London, and Dr Vanessa Didelez, Lecturer in Statistics, University of Bristol, who as my secondary supervisors provided some valuable comments. I would like to thank Dr Richard Chandler, Director of Studies, University College London, for his support with LaTex. I am also thankful to Ms Nadja Leith who provided guidance with the thesis template. I am grateful for the support of Dr Suzanne Evans, Dr Anthony Brooms, and Dr Simon Skene, Birkbeck College, University of London. Finally I thank my examiners Professor Jonathan J Forster and Dr Christian Hennig for many helpful comments and suggestions which greatly helped to improve my thesis.

The initial research theme was first motivated by Dr Paul Brown, Statistics Manager, at Which? who provided me with the impetus for researching this area. He also helped highlight some of the pitfalls in the existing methodology Which? use for brand segmentation. He was very inspirational during the early stages of my research where he provided support and encouragement.

I am grateful for the help and support of my client Ernst & Young and colleagues while working on my thesis. Ernst & Young provided some financial support towards my research.

Last but not least, I am especially grateful to all my family and especially to my parents for their support and for putting up with me in this long journey, made particularly difficult by having to juggle my research with demands from work over the last five years.

# Abstract

Following a review of some traditional methods of clustering, we review the Bayesian nonparametric framework for modelling object attribute differences. We focus on Dirichlet Process (DP) mixture models, in which the observed clusters in any particular data set are not viewed as belonging to a fixed set of clusters but rather as representatives of a latent structure in which clusters belong to one of a potentially infinite number of clusters. As more information about attribute differences is revealed, the number of inferred clusters is allowed to grow. We begin by studying DP mixture models for normal data and show how to adapt one of the most widely used conditional methods for computation to improve sampling efficiency. This scheme is then generalized, followed by an application to discrete data. The DP's dispersion parameter is a critical parameter controlling the number of clusters. We propose a framework for the specification of the hyperparameters for this parameter, using a percentile based method. This research was motivated by the analysis of product trials at the magazine Which?, where brand attributes are usually assessed on a 5-point preference scale by experts or by a random selection of Which? subscribers. We conclude with a simulation study, where we replicate some of the standard trials at Which? and compare the performance of our DP mixture models against various other popular frequentist and Bayesian multiple comparison routines adapted for clustering.

# Contents

# List of Figures

# List of Tables

# Nomenclature

**Acronyms**

**ANOVA** Analysis of Variance (p.3)

**CRP** Chinese Restaurant Process (p.35)

**DBDTMC** Duncan's Bayesian Decision Theoretic Method for Clustering (p.22)

**DD** Dirichlet Distribution (p.32)

**DDs** Dirichlet Distributions (p.73)

**DORO** Dorazio's technique for the $\alpha$ selection in a DPM (p.100)

**DP** Dirichlet Process (Abstract Page)

**DPM** Dirichlet Process Mixture (p.5)

**DPMM** Dirichlet Process Multinomial Mixture (p.73)

**DPMMC** Dirichlet Process Multinomial Mixture model for Clustering (p.77)

**DPNM** Dirichlet Process Normal Mixture (p.43)

**DPNMC** Dirichlet Process Normal Mixture model for Clustering (p.52)

**EDF** Empirical Distribution Function (p.91)

**FDR** False Discovery Rate (p.11)

**FDRC** False Discovery Rate for Clustering (p.21)

**FWER** Familywise Error Rate (p.9)

# Chapter 1

# Introduction

## 1.1 Motivation

The initial motivation for this thesis was driven by some of the problems I faced whilst working as a Statistician at Which? analysing and drawing conclusions from experimental results for publication. Data analysis there often involves the comparison of observed outcomes from two or more brands trials. For example, suppose we have an experiment that compares two brands from the same type of product being assessed on a specific question of interest. Assessors are asked to rate the brands on a preference scale of 1-5, where 1 is poor and 5 good. Usually, in a user trial, the assessors are a random selection of Which? subscribers. Using the data a conventional $t$-test can be used to test for a difference between the population means of the two brands at the conventional 5% level of significance. However, these responses are discrete so assuming normality under the $t$-test is questionable. However, this is common practice currently at Which?.

Although the initial motivation for this thesis was driven by some of the problems faced at Which? we also seek to develop a more general framework for model based clustering that can also be exploited in other areas such as modelling individual differences, in which subjects are assumed to belong to one of a potentially infinite number of clusters, see Navarro et al. (2006).

We present an example of a user trial in the next section followed by the technical details of the Which?'s clustering method and its shortcomings.


Figure 1.1: Blob scale symbols: (top) General set (bottom) Specific to gardening Which?

### 1.1.1 Example Which? user trial: Garden Kneelers

Six brands of garden kneelers were tested by 120 gardening enthusiasts. Each gardener was assigned randomly and anonymously to one of the six kneelers. Each kneeler was rated on a 1-5 preference scale, where 1 is low and 5 is high preference, on various kneeler attributes, e.g. level of comfort, durability etc. Since for each brand the responses are on a discrete 1-5 preference scale, or ordinal, it is common practice at Which? to transform them into a weighted sample mean. For example, if for a particular brand ten gardeners selected preference 4, and the other ten selected preference 5 for level of comfort then the mean would be 4.5. Following this transformation, we cluster the six brand means on level of comfort. Which? currently cluster these brands using the method of Normal scores, which we outline in the next section.

Once the brands have been clustered the researchers are often interested in understanding how the brands can be graded into a class of product on a 1-5 blob scale, where 1 is worst and 5 best quality. The current blob scale symbols used at Which? are shown in Figure 1.1. Ideally the researchers are looking for brands in each of the five blobs to allow for better separation. However, we can sometimes observe all brand means in one cluster, e.g. all garden kneelers are of best quality on level of comfort so we assign a 5 blob score, or a red star, for all brands. Assigning blob scores to brands can be a subjective process, where the cluster solution provided by the statistician is used in conjunction with the researchers' knowledge of the brands market picture to decide on the final scores.

### 1.1.2 Which? methodology

Let $X_{ji}$ denote a response to a question from the $i$th individual for the $j$th brand ($j = 1, \ldots, m, i = 1, \ldots, t$), $\mu$ the overall mean across all brands, $\alpha_j$ the brand effect. Then a possible model for the data can be defined as

$$X_{ji} = \mu + \alpha_j + \epsilon_{ji}, \tag{1.1}$$

where the errors $\epsilon_{ji}$ are iid $N(0, \sigma^2)$. This is the standard one-way analysis of variance (ANOVA) with $t$ randomly selected individuals for each brand $j$. Now if we let $\mu_j = \mu + \alpha_j$ denote the mean for brand $j$ then we construct our test hypothesis as

$$\begin{aligned} H_0 &: \mu_1 = \mu_2 = \cdots = \mu_m = \mu \\ H_A &: \text{At least one pair } (\mu_j, \mu_k), j \neq k, \text{ differs.} \end{aligned}$$

We fit model (1.1) using a statistics package[1] to validate the above hypothesis at the conventional $\alpha = 5\%$ level of significance. Based on the output, we decide whether there is sufficient evidence to reject $H_0$ and conclude that at least one pair $(\mu_j, \mu_k)$ differs significantly. More precisely we reject $H_0$ if

$$\frac{BSS/(m-1)}{ESS/m(t-1)} = \frac{\{m(t-1)\}\sum_{j=1}^{m}(\bar{x}_{j.} - \bar{x}_{..})^2}{(m-1)\sum_{j=1}^{m}\sum_{i=1}^{t}(x_{ji} - \bar{x}_{j.})^2} \geq F^{m-1}_{\{m(t-1)\}}(0.95), \tag{1.2}$$

where $\hat{\mu}_j = \bar{x}_{j.} = \sum_{i=1}^{t} x_{ji}/t$ and $\hat{\mu} = \bar{x}_{..} = \sum_{j=1}^{m}\sum_{i=1}^{t} x_{ji}/mt$. Here $BSS$ is the Between brand Sum of Squares and $ESS$ the Error Sum of Squares. Herein we estimate $\sigma^2$ with $\hat{\sigma}^2 = ESS/(m(t-1))$ unless otherwise stated. The focus now turns to the harder problem of clustering the $\mu_j$. Using the proposal of O'Neill and Wetherill (1971), after a significant one-way ANOVA, discontinuities between $\mu_j$ are found by first ordering their estimates $\hat{\mu}_{(1)} \leq \hat{\mu}_{(2)} \leq \cdots \leq \hat{\mu}_{(m)}$. Under the *null* hypothesis $\hat{\mu}_j \sim N(\mu, \frac{\sigma^2}{t})$. Indeed, if this hypothesis were true then

$$\mathbb{E}[\hat{\mu}_{(j)}] = \mu + \frac{\sigma}{\sqrt{t}} r_{(j)}, \tag{1.3}$$

where the normality of a sample $j$ can be assessed by plotting its order statistic against the order statistics that would be expected from a Normal distribution, or the *normal scores*. We can approximate the $j$th normal score by

$$r_{(j)} = \Phi^{-1}\left(\frac{8j-3}{8m+2}\right), \tag{1.4}$$

[1] SPSS is used currently by the statistics team

where $\Phi^{-1}(p)$ is the $pth$ quantile of the standard normal density. Therefore, if a line of slope $\frac{\hat{\sigma}}{\sqrt{t}}$ and y-intercept $\hat{\mu}$ is drawn on a plot containing all the $\hat{\mu}_{(j)}$ then they should lie close to this line. If not, there are apparent cluster boundaries in the plot. Thus, it may be argued that the adjacent means where boundaries occur divide into more than one cluster. More formally, define the slope between two successive means as

$$q_j = \frac{\hat{\mu}_{(j)} - \hat{\mu}_{(j-1)}}{r_{(j)} - r_{(j-1)}}. \tag{1.5}$$

Then we test whether the observed slope $q_j$ differs from the expected slope $\frac{\sigma}{\sqrt{t}}$. That is, test the the hypothesis

$$\begin{gathered} H_0 : \mathbb{E}[q_j] = \tfrac{\sigma}{\sqrt{t}} \\ \text{against} \\ H_A : \mathbb{E}[q_j] \neq \tfrac{\sigma}{\sqrt{t}}. \end{gathered} \tag{1.6}$$

Currently at Which? they use the rejection criterion

$$T_j = \frac{\sqrt{t}}{3\hat{\sigma}} q_j \geq 1. \tag{1.7}$$

If criterion (1.7) is satisfied then the $m$ means are divided into two clusters, where in one cluster we have $\left(\hat{\mu}_{(1)}, \ldots, \hat{\mu}_{(j-1)}\right)$ and $\left(\hat{\mu}_{(j)}, \ldots, \hat{\mu}_{(m)}\right)$ in the other. Within a defined cluster, we search for further sub clusters in an iterative manner using criterion (1.7) until there are no $T_j \geq 1$ within the sub clusters considered. However, we note a few possible flaws in this methodology, the main one being when we have multiple cases where $T_j \geq 1$ at the first stage of the iterative process (i.e. when we consider all $\hat{\mu}_j$). We currently address this issue by taking the first point of discontinuity $j$ at $\max[T_j]$. Therefore, we are essentially defining a discontinuity amongst the set of other discontinuities as the most significant result. We define this the Method of Normal Scores for Clustering (MNSC). However, it could be argued that a more significant result was observed by chance alone or was an experimental error, and repeating the experiment under the same constraints may yield a less significant result. Adopting a strategy where we take the first discontinuity at $\min[T_j]$ could produce marked differences in the final set of clusters, thereby leading to difficulties in deciding whether the $\hat{\mu}_{(j)}$ in the assigned clusters are by chance or a true reflection of the underlying trend in the brand population. The lack of stability in the final cluster solution that results from this approach could potentially be very damaging for Which? If they have, say, two competing brands in the market place and one is assigned a higher blob score then Which? could potentially be sued by the brand manufacturer with the lower score.

## 1.2 Aims and contributions

With the main difficulty clustering brands highlighted in the previous section we explore clustering and classification more broadly to find an alternative solution.

Since clustering and classification are two of the most fundamental data analysis tools in use today and a very rich and broad area for statistical research, we focus our thesis on two areas. Firstly, since Multiple Comparison Methods (MCMs) are very popular in the the design and analysis of experiments community, we consider popular MCMs and their adaptation to clustering. Clustering methods can be split into model and non-model based. Here we consider recent developments in nonparametric Bayesian analysis with regards to model based clustering using a Dirichlet Process Mixture (DPM) model. Here we assume that the objects, each with some random observations, belong to one of a potentially infinite number of clusters in this model. In the Which? context it could be argued why a model based on an infinite number of clusters is necessary when they ideally require five blob classes at the end. However, since the model is based on an infinite cluster model, it is more adaptive and can uncover new classes that have not been previously observed. In addition, as we shall see later in Chapters 4-6, the DPM provides more flexibility in setting the types of cluster boundaries that are commercially, as well as statistically, meaningful. For example, a mean difference of say 0.01 between two brands in different classes could be statistically meaningful. However they might later be merged into the same class by the researchers, using commercial insight. In situations where there are less than five classes in the data, the brands are allocated into classes based on their sample mean. When more than five classes are present, they are merged down to five. The observed clusters in a particular data set are not viewed as belonging to a fixed set of clusters but rather as representatives of a latent structure. As more information about attribute differences is revealed, the number of inferred clusters is allowed to grow. The Dirichlet process enables the model to uncover new clusters therefore learning from the data. The model allows a priori for an infinite number of clusters. It also avoids the use of computationally intensive Markov Chain Monte Carlo (MCMC) algorithms such as reversible jump MCMC.

We make a number of contributions in this thesis. Firstly we extend the standard DPM structure and then compare this with other adapted clustering methods based on MCMs (Bayesian and frequentist) and K-means using a simulation study that depicts some of the commonly performed trials at Which?. We also adapt one of the

widely used stick-breaking representation of the DP to improve sampling efficiency. Since inferences about the level of clustering can be sensitive to the choice of prior assumed for the dispersion parameter in the DPM, an approach is developed for computing the prior in the presence, or absence, of prior information.

The rest of this thesis is structured as follows. In Chapter 2 we review work on the broader area of MCMs, and adapt some of these MCMs for clustering purposes. We also briefly review frequentist and Bayesian views on multiple comparisons. Chapter 3 presents an introduction to Bayesian nonparametrics. We then provide an introduction to the Dirichlet Process (DP) followed by the DPM. A review of various representations of the DP is provided next. An adaptation of the standard DPM for normal data is made in Chapter 4, followed by a simulation study, depicting common Which? user trials, that compares the clustering performance of DPM with some other adapted clustering methods in Chapter 2. In Chapter 5 we generalise our adapted DPM model to cover non-normal data. We then construct a DPM for multinomial data and assess clustering performance with a simulation study as in Chapter 4. In Chapter 6 we detail some of the difficulties in using conventional ways of setting the prior for the dispersion parameter in the DPM. An approach is then developed for computing this prior in the presence, or absence, of prior information. We repeat the simulation study in Chapter 5 to identify any performance gains using this approach. Finally we present some concluding remarks and directions for future research.

## 1.3 Summary

We have addressed some of the clustering problems we currently face at Which?. The existing clustering method tends to output cluster solutions that are not robust statistically, which potentially gives rise to a brand being assigned the wrong blob score. With this in mind, we explore alternative methods for clustering in the next chapter, where we focus specifically on MCMs and their adaptation to clustering along with other traditional clustering methods such as K-means.

# Chapter 2

# Clustering Methods based on Multiple Comparisons

## 2.1 Introduction

In this chapter we address some standard and non-standard methods for clustering. We first give an introduction on hypothesis testing and then highlight problems with multiple testing. We follow this with a general discussion of popular Multiple Comparison Methods (MCMs) to address some of these issues, then propose a framework where MCMs can be adapted for clustering purposes. We conclude with a general review of clustering methods outside the MCMs community, focusing on model and non-model based clustering methods.

## 2.2 Multiple Comparison Methods

Often statistical analysis involves some form of hypothesis testing. This could be, for example, the brand trials in Section 1.1. For any particular test, the question of interest is simplified into two hypotheses between which we have a choice: the *null* hypothesis, $H_0$, against the alternative hypothesis, $H_A$. Given George Box's famous statement 'all models are wrong but some are useful' it may be simpler, in practice, to interpret a situation having the null hypothesis in mind than a more complex alternative. However, this really depends on the context of the application area. We may decide to act as if the null hypothesis is true until we have sufficient evidence to reject it in favour of the alternative. For example in medical applications a new drug may have potential side effects and unless there is strong evidence to suggest

it is better than placebo it won't be used. So here there is no reason to believe that its true effect is really *exactly* equal to placebo.

We frequently encounter two situations:

1. The experiment has been carried out in an attempt to disprove or reject a particular hypothesis, usually $H_0$. Thus we give it more priority so it cannot be rejected unless the evidence against it is sufficiently strong. For instance, $H_0$: Two brands have the same population means on a given attribute question $H_A$: There is a difference between the two means.

2. If one of the two hypotheses is *simpler* we give it priority so that more complicated theory, as highlighted above, is not adopted unless there is sufficient evidence against the simpler one. For example, it is often simpler to claim that there is no difference between two brands on an attribute question than concluding a difference.

For any particular test we assign a predefined *probability*, usually known as the Type I error $\alpha$. It can be thought of as the probability of falsely rejecting $H_0$ in favour of $H_A$, sometimes referred to as the *false positive*. It is common practice to use probability $\alpha = 0.05$, therefore we accept that one in, say, every twenty such *independent* tests will show a false positive if $H_0$ was true. For instance, if we consider an experiment that involves performing 100 independent tests, we would expect five to be declared as significant if each were performed at $\alpha = 0.05$ under $H_0$. This naturally leads to the *multiple comparison* problem. Our preference here is to control the false positive rate not just for any single test, but also for entire family of tests that makes up our experiment. Before getting deeper into this problem, we need to appreciate the vast amount of literature on this topic including a number of review articles. A good overview on multiple comparisons is provided in the book by Hochberg and Tamhane (1987) and also Hsu (1996). Both are excellent contributions to the field and essential reference manuals. Computer intensive methods to adjust the $p$-values of statistical tests for multiplicity are presented in Westfall and Young (1993).

### 2.2.1 Error Rates

Consider a family of $n$ independent tests, where for each test we have $H_0^i$ vs $H_A^i$, $i = 1, \ldots, n$, with the same value of $\alpha$. Here we refer to $\alpha$ as the *Per-Comparison Error Rate* (PCER), i.e. the probability of incorrectly rejecting each $H_0^i$ that make up the

family. Given all $H_0^i$ are true, it is clear that the the number of false positives $X$ follows a Binomial distribution, $B(\alpha, n)$, where $\alpha$ denotes the probability of *success* and $n$ the number of independent tests. Thus, the probability of, say, $k$ such false positives is

$$P(X = k|\text{all } H_0^i \text{ true}) = \left( \begin{array}{c} n \\ k \end{array} \right) \alpha^k (1-\alpha)^{n-k} \tag{2.1}$$

for all $i$. For large $n$ and small $\alpha$ it can be shown that $X \approx Po(n\alpha)$ under $H_0^i$, $i = i, \ldots, n$.

A more relevant error rate is the *familywise error rate* (FWER) denoted by $\pi$. Simply put, it is the probability of incorrectly rejecting *at least one* of the $H_0^i$ that make up the family. Therefore, by using (2.1) it follows that

$$\pi = P(X \geq 1|\text{all } H_0^i \text{ true}) = 1 - (1-\alpha)^n. \tag{2.2}$$

It is clear from (2.2) that, as the number of tests grows, the probability of observing at least one false positive increases. Intuitively this makes sense; for example, toss a biased coin 100 times, where P[Head]=0.05 and P[Tail]=0.95, then we are almost certain to observe at least one head in those tosses. Several multiple comparison methods that control for FWER exist in the literature, the first being the multiple comparison analysis originally proposed by Fisher (1935), who looked at group means. It is a two-step method: first test the overall null hypothesis that all $k$ group means are equal using ANOVA at significance level $\alpha$. Then, if the null hypothesis of equality is rejected, proceed to test all $\left( \begin{smallmatrix} k \\ 2 \end{smallmatrix} \right)$ pairwise differences between means using separate $t$-tests at PCER $\alpha$. Otherwise, when the overall null hypothesis is accepted we terminate the analysis. This is often known as Fisher's least significant difference (LSD) test. However, the LSD does not protect for FWER. Alternatively, if we wish to fix $\pi$ and solve for the PCER $\alpha$ required for each test then

$$\alpha = 1 - (1-\pi)^{\frac{1}{n}}. \tag{2.3}$$

This is often called the Dunn-Šidák method. Since $1 - (1-\alpha)^n \approx n\alpha$ for small $\alpha$, we obtain the commonly known *Bonferroni* method, by taking

$$\alpha = \pi/n. \tag{2.4}$$

The bound in (2.4) is known as the Bonferroni correction and offer protection against FWER. The Dunn-Šidák correction gives a stronger bound than the Bonferroni correction, because, for $n \geq 1$, $\pi/n \leq 1 - (1-\pi)^{\frac{1}{n}}$. But the Šidák correction

requires the additional condition of independence. In some multiple comparison situations, see Section 2.4, using the Šidák correction is wrong. For example, if we knew that for sample mean differences A-B $> 0$ and B-C $> 0$, then logically we know that A-C $> 0$, so A-C cannot be independent of A-B and B-C.

### 2.2.2 Multiple Comparison Methods

Procedures that are designed to take account of and protect FWER are called MCMs (Multiple Comparison Methods). They can be categorized as either *single-step* or *stepwise*. In operation they differ by the nature in which they take account of decisions on null hypotheses of the same family when testing the actual one. For instance, with single-step methods each null hypothesis is tested without reference to the others in the family. However, in the case of stepwise methods the decisions on already tested hypotheses are used to decide on the rejection or acceptance of another hypothesis. An example of a single-step method has already been presented in the previous section, the Bonferroni test. Other methods that protect FWER include Tukey's procedure for equal sample sizes and the Tukey-Kramer procedure for unequal sample sizes, Dunnett's procedure when population means are compared against a control, Duncan's procedure and procedures based on approximations like those of Bonferroni and Šidák. When population variances are not equal procedures such as Cochran's (C) and Tamhane's (T3) are appropriate.

Contrary to single-step methods, stepwise methods make comparisons in a series of steps, where based on the current step we decide whether to make comparisons in the subsequent step. We can divide stepwise methods into two types: step-up or step-down. The LSD method introduced in the previous section is an example of a step-down procedure, as we only test a subset of means that have been rejected in an earlier step. Other popular step-down procedures are the Newman-Keuls and Duncan multiple range tests. The idea here is to test the observed difference between ordered means, starting with the largest vs smallest, and comparing this to a predefined critical value[1]. Next the difference of the largest and the second-smallest is computed and compared to the critical value. These comparisons are continued until all means have been compared with the largest mean. Then, the difference between the second-largest mean and the smallest is computed and compared. This sequence of comparisons is continued until the difference between all pairs of means have been considered. To prevent contradictions, no differences between a pair of means are considered significant if the two means involved fall between two other

[1]The critical value varies according to the pair of ordered means considered

means that do not differ significantly. The implication is that the procedure could stop early if there are no differences between means at an earlier stage. Thus far we have made statements on whether differences between means are significant, or not, based on a given cut-off value.

Alternatively, we could look at the set of $p$-values to assess the significance of each comparison. One such method that works in this way is the Bonferroni-Holm procedure, Holm (1979). Essentially this is a stepwise version of the Bonferroni test, and proceeds as follows: Order the $p$-values from the $n$ hypotheses, such that $p_{(1)} \leq p_{(2)} \cdots \leq p_{(n)}$. Then, starting with the smallest $p$-value, if $p_{(1)} \leq \pi/n$ the corresponding hypothesis is rejected and the next hypothesis can be tested with $p_{(2)}$. Otherwise, the procedure stops. So in general if we have already rejected $h$ hypothesis then, if in step $i = h+1$ the $p$-value $p_{(h+1)} > \pi/(n-i+1)$ the procedure stops and we accept all remaining hypothesis from $i = h+1, \ldots, n$. A variant of this is the Simes-Hochberg approach, Simes (1986). As opposed to stopping when we fail to reject a hypothesis, we start backwards working with the largest $p$-values first. That is, if $p_{(n)} > \pi$ the corresponding hypothesis is accepted, then we test the next hypothesis corresponding to $p_{(n-1)}$. Therefore, if we have already accepted $h$ hypothesis then, if at step $i = h+1$ the $p$-value $p_{(n-i+1)} \leq \pi/i$, the procedure stops and we reject all remaining hypotheses from $i = h+1, \ldots, n$. Although the Simes-Hochberg approach is more powerful than Holms, it is only strictly applicable when the tests within a family of hypothesis are independent, whereas Holms approach does not have this restriction.

If we had a situation where there were a large number of null hypotheses, then a powerful result based on the distribution of $p$-values can be employed. One creative use is the proposal by Schweder and Spjtvoll (1982), where the idea is to form a $QQ$-plot of $p$-values and look for an elbow separating a linear region coming from a true null hypothesis from that of a false null hypothesis.

We briefly mention step-up procedures, where the idea is to start by testing a single hypothesis. Then, depending on the result, we either step-up to a hypothesis involving more means or stop. The literature on step-up procedures is rather limited. However Welsch (1977) addressed them in some detail. More recent proposals can be found in Hochberg (1988) and Dunnett and Tamhane (1992).

### 2.2.3 False Discovery Rate

Benjamini and Hochberg (1995) have been particularly influential in multiple comparisons, in particular their proposal of the false discovery rate (FDR) control as an

| | Declared significant | Declared not significant | Total |
|---|---|---|---|
| Null true | $F$ | $n_0 - F$ | $n_0$ |
| Alternative true | $T$ | $n_1 - T$ | $n_1$ |
| Total | $S$ | $n - S$ | $n$ |

Table 2.1: Possible outcomes from $n$ hypothesis tests based on a significance rule

alternative to the commonly used FWER, see Section 2.2.1. The FDR is the fraction of false positives among all tests declared significant. The motivation for using the FDR is that we may be running a very large number of tests, with those being declared significant being subjected to further studies. Examples range from a large scale application: differential expression over a huge set of genes on a microarray, to a small scale application, see Section 1.1. Fewer applications have been proposed when the data are discrete in nature. However, a recent proposal by Gilbert (2005) looking at human immunodeficiency virus data developed a modified FDR procedure that is more powerful under this setting.

The initial analysis takes a large number of candidates and produces a smaller subset for further analysis. Therefore, we are more concerned with making sure all possible true alternatives are included in this subset, and we are willing to put up with some false positives to accomplish this. However, we also do not want too many false positives. Therefore, we need to define the FDR rate $\delta$, where we expect that a proportion $\delta$ of candidates in the subset declared as significant are actually false positives. Conversely a proportion $1-\delta$ of those candidates declared significant will be the correct decision. Usually $\delta$ is taken to be 0.05, which is also the usual rate for FWER.

To formally motivate the FDR, suppose a total of $n$ hypotheses are tested, $S$ of which are judged significant[1]. If we had complete knowledge then we would know that $n_0$ of the null hypotheses were true and the remaining $n_1 = n - n_0$ null hypotheses were false. We might find that a number $F$ of the true nulls is declared significant while a number $T$ is declared significant when the alternative is true. For clarity we illustrate this in Table 2.1. From Table 2.1 it follows that FDR $\delta = F/S$. In contrast, note that $\alpha = \mathbb{E}[F]/n_0$. In both cases notice the denominators are considerably different. In the first case the number of hypotheses, $S$, that are declared significant, whereas the number, $n_0$, that are truly null in the second. Another way to see the distinction between $\alpha$ and $\delta$ is to consider them as

[1]based on some criterion used for each test

| Quantity | Definition |
|---|---|
| $\alpha$ | Comparison-Wise Type I error (false positive) |
| $\beta$ | Type II error (false negative), where $1-\beta$=Power |
| $\pi$ | Family-wise Type I error, Pr($F>0$)=$\pi$ |
| $\delta$ | False Discovery Rate |
| $\pi_0$ | Fraction of all hypotheses that are null |
| $T$ | Test statistic |

Table 2.2: Notation for multiple comparisons

probability statements on a single hypothesis $i$. Then

$$\delta = \mathrm{P}(i \text{ is truly null} | i \text{ is significant}), \tag{2.5}$$

whereas

$$\alpha = P(i \text{ is significant} | i \text{ is truly null}). \tag{2.6}$$

Now, let us remind ourselves as to the various parameters that arise when multiple comparisons are considered, see Table 2.2. More importantly, we are interested in how these parameters are related to each other. First, to understand the relationship between $\alpha$, $\pi$, and $F$, let us consider the situation where we set the false positive rate at $\alpha$. Then, given $n$ tests under the null ($p \leq \alpha$ is classed as significant and a false positive) the expected number of false positives under the null is bounded above by $\mathbb{E}[F] = n\alpha$. We now consider the relationship between $\alpha$, $\beta$, $\pi_0$, and $\delta$. However, to do so we first need to consider the concept of the posterior error rate (PER). The PER was first introduced in the context of linkage analysis in humans by Morton (1955). Simply put, Morton's PER is the probability when $n = 1$ that a single significant test is a false positive

$$PER = P(F = 1 | S = 1). \tag{2.7}$$

If we base tests on PER then we encounter the *screening paradox* noted by Manly et al. (2004): Type I error control may not lead to a suitably low PER. For example, we might set $\alpha = 0.05$, which may result in the PER being much higher. Therefore the tests being significant may have a much higher false-positive rate than 5%. The key distinction here is to observe that, rather than conditioning on the hypothesis being null as we do with $\alpha$, we condition on the test being significant. Therefore, in the pool of significant tests, we could either have false or true positives. The relative fraction of each is a function of $\alpha$, $\beta$ and $\pi_0$. To see this more clearly, we

apply Bayes' theorem to (2.7) to give

$$PER = P(F = 1|S = 1) = \frac{P(S = 1|H_0 = \text{True})P(H_0 = \text{True})}{P(S = 1)}. \tag{2.8}$$

If we denote the fraction of all $n$ hypotheses that are truly null by $\pi_0 = n_0/n$, then

$$P(S = 1|H_0 = \text{True})P(H_0 = \text{True}) = \alpha\pi_0. \tag{2.9}$$

Now, considering the denominator of (2.8), we need to work out the probability that a single randomly drawn test is declared significant. This can occur if we pick a null hypothesis as significant, with probability $\alpha$, or if we pick an alternative hypothesis and avoid a Type II error $\beta$. Therefore

$$P(S = 1) = \alpha\pi_0 + (1 - \beta)(1 - \pi_0). \tag{2.10}$$

Thus (2.8) reduces to

$$\begin{aligned} PER = P(F = 1|S = 1) &= \frac{\alpha\pi_0}{\alpha\pi_0 + (1 - \beta)(1 - \pi_0)} \\ &= \frac{1}{1 + \frac{(1-\beta)(1-\pi_0)}{\alpha\pi_0}}. \end{aligned} \tag{2.11}$$

We note that when $\pi_0$ is close to 1 most hypothesis are null. However, more realistically, as some of the hypotheses are expected not to be null (1-$\pi_0$ is modest to large).

While the FDR for any given experiment is simply $F/S$, there are several ways in which we could formally define the expectation of this ratio. The original notion of FDR was suggested by Benjamini and Hochberg (1995), defined as

$$\text{FDR} = \mathbb{E}\left[\frac{F}{S}\middle| S > 0\right] P(S > 0). \tag{2.12}$$

Since then a number of workers have suggested modifications, the most distinct contributions from Storey (2002): the positive false discovery rate

$$pFDR = \mathbb{E}\left[\frac{F}{S}\middle| S > 0\right]. \tag{2.13}$$

We condition on $S > 0$ to allow for cases when $S = 0$. Another important contri-

bution is the proportion of false positives

$$PFP = \frac{\mathbb{E}[F]}{\mathbb{E}[S]} \tag{2.14}$$

defined by Fernando et al. (2004). Others include PER as described before and the False Positive Rate FPR=P($F \geq 1$).

Strictly speaking, these are the proportion of false positives. This is a good thing, as Fernando et al. (2004) have shown that the PFP does not depend on either the number of tests or the correlation structure among tests (essentially this occurs because we are taking the ratio of two expectations, so the number of tests cancels in each and correlation structure among tests does not enter into the individual expectations). In essence the main operational differences between the different false discovery rates are

1. The original method by Benjamini and Hochberg (1995) assumes $n = n_0$ (all hypotheses are true nulls)

2. All other estimators assume $\pi_0$ is not necessarily one, thus attempt to estimate $\pi_0$ or $n_0$ and then use these to estimate the corresponding false discovery rate.

While we can control the FDR for an entire set of experiments, we would also like to have an indication of the FDR for any particular experiment (or test) within this family. Intuitively, tests with smaller $p$-values should also have smaller associated FDR values. Storey (2002), and Storey (2003) introduced the concept of a $q$-value (as an alternative to $p$-value) of any particular test, where $q$ is the expected FDR rate for tests with $p$-values at least as extreme as the test of interest. The estimated $q$-value is a function of the $p$-value for that test and the distribution of the entire set of $p$-values from the family of tests being considered.

The difficulty is now in estimating $\pi_0$, the proportion of true null hypotheses. We consider the distribution of $p$-values under the null being uniform. If some alternative hypotheses are true then they are mixed in with the null hypotheses. Therefore, we expect the distribution of $p$-values to be a mixture, with $n_0/n$ draws from a uniform and $(1-n_0)/n$ draws from some other distribution in which the $p$-values are skewed towards zero. The main offerings can be summarised as follows: first Schweder and Spjtvoll (1982) make use of a regression estimator to estimate $\pi_0$; however this tends to overestimate the number of nulls. Another approach was suggested by Allison et al. (2002), who used maximum likelihood (ML) to fit a mixture model to the $p$-values. Finally, a very simple estimator was offered by Storey (2003), using the key feature that draws from hypotheses which are not null are expected to have their

$p$-values skewed towards zero. Although current methods for the estimation of $\pi_0$ provide adequate results in many situations, it was pointed out by Black (2004) that when the data arise from mixtures of distributions which are difficult to separate, the development of improved estimation techniques will allow better control of error.

Another area of research is in the development of FDR controlling techniques for dependent hypothesis tests. There have been relatively few advances in this area. Nonetheless, the most marked contributions have come from Storey et al. (2004), who considered a form of weak dependence under which the distribution function of both null and non-null $p$-values approach limit functions. He then went on to show the asymptotic control of FDR in this case. An approach based on a permutation procedure was proposed by Korn et al. (2004) where the idea was to fix the probability of a given number of false positives below $\alpha$. Further, to highlight the inflation of variance of the false discoveries, Owen (2005) presents a variance formula to take account of correlations between test statistics. Other recent advances include the recent proposal by Wenguang and Tony (2009) where they tackle the dependence using a hidden Markov model. Development of FDR controlling multiple-comparison techniques is an active area of research, and we expected that many of the newly developed procedures will build on the fundamentals proposed by Benjamini and Hochberg (1995).

### 2.2.4 Bayesian views on Multiple Comparisons

Inconsistencies with the scientific method and the likelihood principle have been the common complaint with the frequentist approach to hypothesis testing, see Berry (1988), Berger and Berry (1988) and Berger and Wolpert (1984). For instance, suppose we are interested in testing $\theta$, the unknown probability of heads for a possibly biased coin. Suppose, $H_0 : \theta = 0.5$ vs $H_a : \theta > 0.5$. An experiment is conducted and nine heads and three tails are observed in twelve flips of a coin. This information is not sufficient to fully specify the $p$-value, since when the number of flips is fixed at $n = 12$ we have the number of heads $X \sim B(n, \theta)$, from which it follows that $P[X \geq 9|H_0 = True] = 0.073$ so we accept $H_0$ at the $\alpha = 0.05$ level of significance. However, if we decided to flip until the third time a tail is observed then the number of heads, $X$, before the third tail appears is Negative Binomial (NB), where $X \sim NB(3, 1-\theta)$. Here we find that $P[X \geq 9|H_0 = True] = 0.033$ so we reject $H_0$. However the likelihood function is $f(x|\theta) \propto \theta^9(1-\theta)^3$ in each case. This inconsistency of $p$-values violates the likelihood principle.

As we have seen in Section 2.2.1, the frequentist approach to multiple compar-

isons rests primarily on controlling the FWER. However, these tests often tend to be conservative, especially when we have a larger number of tests. They have been criticised for paying too much in terms of power for achieving the desired level of FWER control. Therefore procedures that try to overcome these difficulties within the frequentist approach are the subject of current research within the area.

It was shown by Westfall and Johnson (1997) that Bayesians will come close to either the PCER or FWER depending on the credibility they attach to the family of null hypotheses under consideration when using a single-step MCM in the context of ANOVA.

The first fully Bayesian approach to the multiple comparison problem was by Duncan (1965). In this work he outlined the problem of pairwise comparisons in a one-way layout, a decision-theoretic approach assuming additive losses produces the usual comparisonwise approach. One of Duncan's achievements was to shed new light on the problem of multiple comparisons by using a decision-theoretic based approach. Here, following the derivation of the posterior distribution for the relevant parameters, the next step involves some decision analysis. Therefore, considering two or more means to be equal under the Bayesian framework, involves considering the impact of various decisions explicitly in terms of loss functions. Another achievement by Duncan was to break the ice between frequentists who thought that Bayesians had nothing to contribute to the multiple comparisons problem, and Bayesians who found no reason to adjust for multiple comparisons.

An extension of the original Duncan's procedure was proposed by Shaffer (1999), where rather than controlling Type I error, she replaces this by controlling the seriousness of Type I and Type II errors using linear loss functions. This is basically a modification of the formulation provided by Waller and Duncan (1969), which is based on the original Bayesian procedure of Duncan (1965).

One of the advantages of Bayesian MCMs is that they allow for direct probability calculations of the hypotheses of equality and inequality of means. However, the specification of prior probabilities for the hypotheses concerned can be seen as a possible hurdle. In recent years we have seen remarkable developments in the area of Bayesian nonparametric inference both from a theoretical and applied perspective. As for the latter, the celebrated Dirichlet process has been successfully exploited within Bayesian mixture models leading to many interesting applications, such as multiple comparisons, see Berry (1988). As for the former, some new discrete nonparametric priors have been recently proposed in the literature: their natural use is as alternatives to the Dirichlet process in a Bayesian hierarchical model for density estimation, see Escobar and West (1995). When using such models for concrete ap-

plications, it could be desirable to investigation their statistical properties. Among them a prominent role is to be assigned to consistency. Indeed, strong consistency of Bayesian nonparametric procedures for density estimation has been the focus of a considerable amount of research and, in particular, much attention has been devoted to the Dirichlet process normal mixtures, see Ishwaran and James (2002).

In the next section we consider a few MCMs that we will adapt for clustering later.

## 2.3 Other MCMs

We now consider a few MCMs outlined by Shaffer (1999), where part of her study involved comparing various Bayesian and non-Bayesian procedures under frequentist concepts, namely *power* and FWER. We summarise her setup. Let $X_{ji} \sim N(\mu_j, \sigma^2)$, $i = 1, \ldots, t$, $j = 1, \ldots, m$, so that $\bar{X}_j \sim N(\mu_j, \sigma^2/t)$, where $\bar{X}_j = \sum_{i=1}^{t} X_{ji}/t$ and we assume $\sigma^2$ is known. Under independence, it follows that $\delta_{jk} = \mu_j - \mu_k$, and $D_{jk} = \bar{X}_j - \bar{X}_k \sim N(\delta_{jk}, 2\sigma^2/t)$ respectively, $1 \leq j < k \leq m$, where we have $n = m(m-1)/2$ $\delta_{jk}$ pairs. In this chapter we let the observed values of $X_{ji}$ be denoted by $x_{ji}$, $\bar{X}_j$ by $\bar{x}_j$ and $D_{jk}$ by $d_{jk}$. Next, the $d_{jk}$'s are ordered from smallest to largest and subscripts matched with $\delta_{jk}$. For each difference $\delta_{jk}$, we are interested in three hypotheses, namely

$$H_{jk1}: \quad \delta_{jk} < 0 \qquad H_{jk2}: \quad \delta_{jk} > 0 \qquad H_{jk}: \quad \delta_{jk} = 0.$$

Thus we have three possible decisions: Reject $H_{jk1}$ and decide $\delta_{jk} \geq 0$, reject $H_{jk2}$ and decide $\delta_{jk} \leq 0$, or reject $H_{jk}$ and decide $\delta_{jk} \neq 0$.

We now summarise the non-Bayesian procedures that were compared with Shaffer's modification of Duncan's procedure as follows:

1. The conventional $z$-test, assuming $\sigma^2$ is known, is used to reject $H_{jk}$ when

$$|d_{jk}| > \sigma\sqrt{\frac{2}{t}} Z_{\frac{\alpha}{2}}, \tag{2.15}$$

where $Z_{\frac{\alpha}{2}}$ is the upper $\frac{\alpha}{2}$ critical value of the standard normal distribution. If $H_{jk}$ is rejected we decide $\delta_{jk} \geq 0$ if $d_{jk} > 0$ else $\delta_{jk} \leq 0$ if $d_{jk} < 0$. This procedure is designated SEP since the hypotheses are treated separately and don't control for FWER or the FDR. Notice here that no control for multiple comparisons is made as each of the $n$ hypotheses is tested separately without regards to the increase in Type I error.

2. RANGE is a single-stage procedure based on the distribution of the range, see Benjamini and Braun (2002). We reject $H_{jk}$ if

$$|d_{jk}| > \frac{\sigma}{\sqrt{t}} q_{m,\pi}, \tag{2.16}$$

where $q_{m,\pi}$ is the upper $\pi$ critical value of the range of $m$ standard normal random variables. We make a decision based on the sign of $d_{jk}$ as with SEP.

3. Using a FDR-controlling procedure in its simplest form, see Section 2.2.3, we reject $H_{jk}$ and decide $\delta_{jk} \neq 0$ based on $p_{jk}$, which is the significance probability of $|d_{jk}|$. Next, the $p_{jk}$ are ordered from smallest to largest, and then we reject all $H_{jk}$ for which $j \leq l$, where $l$ is the largest subscript $j$ for which $p_{jk} \leq j\delta/n$. If no such $l$ exists then we accept all $H_{jk}$. In a similar manner to SEP, amongst the rejected $H_{jk}$ we make a decision based on the sign of $d_{jk}$. This procedure is designated FDR1 to distinguish it from other FDR-controlling procedures.

Finally we consider Shaffer's modification of Duncan's procedure, named DUB. Until now, we have assumed that $\mu_j$ have arbitrarily fixed values. However, Duncan assumes that $\mu_j \sim N(0, \tau^2)$. Then by construction of loss functions across all possible decisions we select the one that minimises the expected loss. More formally, if we let $\theta = (\mu_1, \ldots, \mu_m)$, then Duncan defines the loss functions as

$\xi_1$: Decide $H_{jk1}$

$$L(\theta, \xi_1) = \begin{cases} (k_1 + k_2)\delta_{jk} & ; \delta_{jk} \geq 0 \\ 0 & ; \delta_{jk} < 0 \end{cases} \tag{2.17}$$

$\xi_2$: Decide $H_{jk2}$

$$L(\theta, \xi_2) = \begin{cases} -(k_1 + k_2)\delta_{jk} & ; \delta_{jk} \leq 0 \\ 0 & ; \delta_{jk} > 0 \end{cases} \tag{2.18}$$

$\xi_3$: Decide $H_{jk}$

$$L(\theta, \xi_3) = \begin{cases} k_2 \left|\delta_{jk}\right| & ; \delta_{jk} \neq 0 \\ 0 & ; \delta_{jk} = 0 \end{cases} \tag{2.19}$$

The ratio $k^* = k_1/k_2$, can be thought of as the ratio of the loss due to a Type I error, $k_1$, to the loss due to a Type II error, $k_2$, in testing a single directional hypothesis. Instead of being fixed as in Duncan's formulation, Shaffer (1999) chooses $k^*$ such that the FWER is $\pi$ in the complete null case. It was shown by Shaffer (1999) that

with the DUB method we reject $H_{jk}$ if

$$|d_{jk}| > \sigma\sqrt{\frac{\Psi}{t(\Psi-1)}}t_{\infty}, \tag{2.20}$$

where

$$\Psi = \mathbb{E}[\text{MSB}]/\mathbb{E}[\text{MSW}]. \tag{2.21}$$

Here MSB and MSW are the between-group and the within-group mean squares, respectively, in a one-way layout analysis of variance. Also $t_{\infty}$ is the value of $z$ for which the risk ratio

$$k^* = \frac{\phi(z) + z\Phi(z)}{\phi(-z) - z\Phi(-z)}, \tag{2.22}$$

where $\phi$ and $\Phi$ are the standard normal density and cumulative distribution functions respectively. In the special case when $\Psi = 1$ we accept all hypotheses, also note that the RHS of (2.20) can potentially be negative when $\Psi - 1 < 0$.

## 2.4 Adaptation of MCMs for Clustering

In this section we adapt the MCMs introduced in the last section for clustering purposes. We first start with the adaptation of RANGE to the Tukey's Method for Clustering (TMC). Consider a set of population means $\mu_j$, with corresponding sample means as defined in the last section. The mechanics for the clustering follows in a step-down fashion, but first we order $\bar{x}_{(1)} \leq \cdots \leq \bar{x}_{(m)}$ then we proceed as follows:

1. If $\left(\bar{x}_{(l)} - \bar{x}_{(1)}\right) \geq C_{l,1}(\gamma)\ \forall l \in \{m, m-1, \ldots, 2\}$ is satisfied, we reject $H_{l1}$ and a cluster boundary is placed between $\bar{x}_{(1)}$ and $\bar{x}_{(2)}$, thus separating the means into two clusters, one that contains $\bar{x}_{(1)}$ and $\left\{\bar{x}_{(2)}, \ldots, \bar{x}_{(m)}\right\}$ in the other. We carry on to the next step even if we do not reject $H_{l1}$.

2. Next if $\left(\bar{x}_{(l)} - \bar{x}_{(2)}\right) \geq C_{l,2}(\gamma)\ \forall l \in \{m, m-1, \ldots, 3\}$ is satisfied, we reject $H_{l2}$ and a cluster boundary is placed between $\bar{x}_{(2)}$ and $\bar{x}_{(3)}$, therefore if $H_{l1}$ was rejected in the previous step, separating the relevant means further into two clusters, one that contains $\bar{x}_{(2)}$ and $\left\{\bar{x}_{(3)}, \ldots, \bar{x}_{(m)}\right\}$ in the other.

3. We continue until we reach the inequality $\left(\bar{x}_{(m)} - \bar{x}_{(m-1)}\right) \geq C_{m,m-1}(\gamma)$. If satisfied we reject $H_{m(m-1)}$, and assuming all $(H_{l1}, \ldots, H_{(m-1)(m-2)})$ were rejected previously, we put a cluster boundary between $\bar{x}_{(m-1)}$ and $\bar{x}_{(m)}$ therefore

separating $\bar{x}_{(m-1)}$ and $\bar{x}_{(m)}$ in the final two clusters[1][2][3].

Here $C_{k,j}(\gamma)$ is the critical value that is used for rejection between the relevant pair of means, where $\gamma$ is a vector of parameters used in the proposed method.

Specifically, if we consider TMC, we see that

$$C_{k,j}(\pi) = q_{\pi}(m, m(t-1))S.E.M,$$

where $q_{\pi}(m, m(t-1))$ is the upper $\pi$ percentage point of the studentized range from $m$ means and $m(t-1)$ error degrees of freedom. The standard error, when we have a fixed sample size $t$, is

$$S.E.M = \sqrt{\frac{\sum_{k=1}^{m} s_j^2}{mt}}.$$

Here $s_j^2 = \sum_{i=1}^{t}(x_{ji} - \bar{x}_j)^2/(t-1)$.

Next, with the False Discovery Rate for Clustering (FDRC), $C_{k,j}$ is binary where 1 signifies reject and 0 accept. We determine $C_{k,j}$ by first computing

$$p_{kj} = 2\left[1 - \Phi\left\{\frac{\sqrt{t}(\bar{x}_k - \bar{x}_j)}{\sqrt{2\sigma^2}}\right\}\right],$$

where $\sigma^2$ is estimated by the usual pooled estimate of variance $\hat{\sigma}^2 = \sum_{j=1}^{m} s_j^2/m$ when we have a fixed sample size $t$. Then all $m(m-1)/2$ $p_{kj}$ are ordered from from the smallest to the largest. We denote the ordered $p_{kj}$ by $p_{(q)}$, where $q = 1, \ldots, m(m-1)/2$. Let $h$ be the largest subscript for which $p_{(q)} \leq 2q\delta/m(m-1)$. If no such subscript exists we reject no corresponding hypothesis associated with $p_{kj}$, therefore all $C_{k,j} = 0$. Otherwise, we reject all corresponding hypotheses for $q \leq h$ and accept for $q > h$. Then for the corresponding $p_{kj}$ of the rejected hypotheses we set $C_{k,j} = 1$, otherwise $C_{k,j} = 0$ for the corresponding $p_{k,j}$ of the accepted hypotheses. We then adapt for clustering as follows:

1. If $C_{l,1} = 1 \ \forall l \in \{m, m-1, \ldots, 2\}$ is satisfied, we reject $H_{l1}$ and a cluster boundary is placed between $\bar{x}_{(1)}$ and $\bar{x}_{(2)}$, thus separating the means into two

[1] In total for $m$ means we make $\frac{m(m-1)}{2}$ comparisons.

[2] In constructing this clustering technique we have ignored the dependence between sample means.

[3] Sometimes, under this technique, clusters may contain sample means that are significantly different. To illustrate this consider three ordered sample means from brands A, B and C respectively. Brand A could be significantly different to C, but neither A nor C are significantly different from B between them. This is not necessarily a problem in the Which? context, see Section 1.1.1, as they ideally seek five clusters. When more than five clusters are observed they are usually merged down to five using commercial insight.

clusters, one that contains $\bar{x}_{(1)}$ and $\left\{\bar{x}_{(2)}, \ldots, \bar{x}_{(m)}\right\}$ in the other. We carry on to the next step even if we do not reject $H_{l1}$.

2. Next if $C_{l,2} = 1 \ \forall l \in \{m, m-1, \ldots, 3\}$ is satisfied, we reject $H_{l2}$ and a cluster boundary is placed between $\bar{x}_{(2)}$ and $\bar{x}_{(3)}$, therefore if $H_{l1}$ was rejected in the previous step, separating the relevant means further into two clusters, one that contains $\bar{x}_{(2)}$ and $\left\{\bar{x}_{(3)}, \ldots, \bar{x}_{(m)}\right\}$ in the other.

3. We continue until we reach $C_{m,m-1} = 1$. If satisfied we reject $H_{m(m-1)}$, and assuming all $(H_{l1}, \ldots, H_{(m-1)(m-2)})$ were rejected previously, we put a cluster boundary between $\bar{x}_{(m-1)}$ and $\bar{x}_{(m)}$ therefore separating $\bar{x}_{(m-1)}$ and $\bar{x}_{(m)}$ in the final two clusters.

When adapting the DUB to Duncan's Bayesian Decision Theoretic Method for Clustering (DBDTMC), we assume $\mu_j \sim N(0, \tau^2)$, whereas with the other methods we assumed them to be fixed. Then the posterior distribution of $\mu_j$ is

$$\mu_j | \bar{x}_j \sim N\left(\frac{\frac{t\bar{x}_j}{\sigma^2}}{\frac{t}{\sigma^2} + \frac{1}{\tau^2}}, \frac{1}{\frac{t}{\sigma^2} + \frac{1}{\tau^2}}\right).$$

Under a Bayesian decision rule, we choose $\xi$ such that $\mathbb{E}[L(\theta, \xi)|\underline{X}]$ is a minimum, where $\underline{X} = (\underline{X}_1, \underline{X}_2, \cdots, \underline{X}_m)$. We estimate $\tau^2$ empirically from the data[1] by

$$\hat{\tau^2} = \frac{\sum_{j=1}^{m}(\bar{x}_{j.} - \bar{x}_{..})^2}{m-1} - \frac{\sum_{j=1}^{m} s_j^2}{mt},$$

where $\bar{x}_{..} = \sum_{j=1}^{m}\sum_{i=1}^{t} x_{ji}/mt$. As with FDRC, $C_{k,j}$ is also binary so our clustering method can be applied as before. However, unlike FDRS, the relevant cluster boundary is placed when we decide $\xi_3$, where we reject $H_{jk}$.

Finally, considering the K-means, see next section, for Clustering we relabel this KMeansC. With KMeansC we simply ran the `kmeans(...)` function in R on the ordered means with the number of clusters prespecified[2]. We also considered index G1 for clustering (G1C), see Gordon (1999), where K-means is simulated with an index to determine the number of clusters. The cluster boundaries resulting from these methods are then constructed.

Note that it is implicitly assumed with MNSC, TMC, and FDRC that objects placed in the same cluster have sample means from the same underlying distribution.

[1] Empirical Bayes

[2] This causes a few difficulties later when assessing KMeansC's performance in relation to the other methods. We address this issue using the third performance measure described in Section 4.4

So here how well these methods detect the *true* number of clusters is dependent on how well they differentiate between cases where two, or more, distributions are put together in the same cluster unless each object's sample size $t$ is large enough that significant differences can be found. With DBDTMC a cluster is defined as a set of objects where any pair has minimum posterior expected loss under decision $\xi_3$, see 2.19. When 2.19 is decided the corresponding pair or objects are put in the same cluster. The truth here is determined by how well the method differentiates between decision $\xi_3$ and the others. Finally, the underlying truth for KMeansC and G1C is defined in the next section. Underlying all these methods is the insight that in no situation can it be clear what the true clusters are from the data alone, and extra information is needed, for instance, from the researchers in our Which? example in Section 1.1.1. Later, in the simulation studies of Chapters 4-6 we compare the performance of these methods with two others based on the DPM.

## 2.5 General Clustering

Thus far we have only considered clustering based on MCMs, but clustering can be thought of more broadly. We start by stating the basic clustering problem simply. Given a set of $n$ distinguishable objects, we wish to distribute the objects into clusters in such a way that the objects within a cluster are similar, whereas the clusters themselves are different. Cluster analysis is a set of statistical methods that cluster individual observations into classes, called clusters, on the basis of similarity. Many clustering algorithms have been proposed in various fields, see Hartigan (1975). Of this set, the two most common non-model-based clustering methods applied in standard settings are hierarchical and K-means cluster analysis, see MacQueen (1967).

Cluster analysis techniques can be broadly separated into two approaches, hierarchical and nonhierarchical. The hierarchical approach builds clusters of successively larger size using some measure of similarity or distance.

Hierarchical cluster analysis (HCA) comprises two separate methods, agglomerative and divisive. When using hierarchical agglomerative clustering, each individual observation is initially designated as a separate cluster. In a stepwise fashion, the most similar clusters are combined into larger units, ending when there exists one super-cluster containing all observations. In contrast, the divisive technique begins with the single super-cluster, and proceeds stepwise by dividing the cluster into its most dissimilar two parts. This process repeats, ending when there are $n$ clusters, one for each observation. Hierarchical clustering can be used in standard settings to define a set of cluster solutions and each solution can then be evaluated for its

respective fit of the data. Typical algorithms used in this approach include single linkage (nearest neighbour), complete linkage (furthest neighbour), and Wards method, which minimizes the mean square distance between the centre of a cluster and each member.

Nonhierarchical clustering approaches also exist, including the K-means method. K-means cluster analysis starts with the user identifying the number of clusters desired, and is based on the Euclidean distance by definition. An individual observation is compared with the values of each centroid and assigned to the cluster with which it is most similar. The value of each affected centroid is recalculated after each new assignment. The process is complete when, after a complete pass through the dataset, no re-assignments are made. The main advantages of this method are its simplicity and speed which allows it to run on large datasets. However, due to the initial random assignments of the centroids, it doesn't always yield the same result with each run. One of the restrictions with the standard K-means is that the number of clusters have to be prespecified. Instead one could use index G1, see Gordon (1999, p.61), which is a combination of K-means with an index. Here K-Means is run for each of the $[2, (n-1)]$ cluster solutions. The solution that maximizes the ratio of the between and within cluster variance is taken as a the final. One of the drawbacks with these methods is that they can't handle the one, or $n$, cluster solution. However, we can use the Duda and Hart's criterion L1, see Gordon (1999, p.62), to compare the one and two cluster solutions. Since K-means can actually be linked to a classification model based on several spherical normal populations with the same variance, this can be seen as the underlying truth, see Gordon (1999, pp.65-68). With G1C the true number of clusters is to be estimated from this underlying model.

Both of the above methods of cluster analysis use similarity between observations as the basis of categorization. Since all data can be represented as vectors (one-dimensional arrays) similarity is defined geometrically. Although several alternatives exist for defining this similarity, the most commonly used is *Euclidian Distance*. The Euclidean distance between points $P = (p_1, \ldots, p_n)$ and $Q = (q_1, \ldots, q_n)$ is defined as $\sqrt{\sum_{s=1}^{n}(p_s - q_s)^2}$. Another commonly used measure of Euclidean distance that does directly incorporate a standardisation procedure is the *Mahalanobis distance*. The Mahalanobis approach not only incorporates a standardisation process on the data, but also adjusts the intercorrelations among the variables[1].

Both HCA and K-means cluster analysis can produce many solutions for a given problem. For example, HCA produces a set of cluster solutions whose number

[1]This distance measure is not the standard measure of distance with K-means

equals the number of elements clustered. Therefore, some criteria must be available to provide selective support for some cluster solutions over others.

Thus far we have considered non-model based clustering methods. Alternatively we can also base a clustering algorithm on the assumption that the measurements to be clustered are realizations of a random vector from some parametric statistical model. More precisely, in model-based clustering it is assumed that the data are generated by a mixture of underlying probability distributions in which each component represents a different cluster. The mixture proportions sum to one across the number of mixtures considered. This distribution is commonly Gaussian, a model that has been used with considerable success in a number of applications, see Banfield and Raftery (1993). In a classical framework we use the Expectation-maximization (EM) algorithm for finding maximum likelihood estimates of parameters in models, see Dempster et al. (1977). In standard nonhierarchical cluster techniques, the allocation of objects to clusters should be optimal according to some criterion. These criteria typically involve minimizing the within-cluster variation and/or maximizing the between-cluster variation. An advantage of using a statistical model is that the choice of the cluster criterion is less arbitrary. Nevertheless, the criteria that arise from a log-likelihood analysis of model based cluster models may be similar to the criteria used by certain nonhierarchical cluster techniques like K-means. An advantage of the model-based clustering approach is that no decisions have to be made about the scaling of the observed variables. For instance, when working with Gaussian distributions with unknown variances the results are the same irrespective of whether the variables are normalized or not. This differs from the standard non-hierarchical cluster methods, where scaling is always an issue. Another advantage is that it is relatively easy to deal with variables of mixed measurement levels. Moreover, we obtain a formal measure of uncertainty for the assignment of each object via the probabilities of cluster membership. However, with mixture models, an identifiability problem arises from the invariance of the likelihood under permutation of the component labels unless strong prior information is used, see Stephens (2000). Traditional approaches to this problem impose identifiability constraints on model parameters. However, these constraints do not always solve the problem. Other solutions can be found in Jasra et al. (2005) who categorize them as artificial identifiability constraints, Green and Richardson (1997), random permutation sampling, Frühwirth-Schnatter. (2001), relabeling algorithms, Stephens (2000), and label invariant loss functions methods, see Celeux et al. (2000). The identifiability problem is not worse, in principle, with mixture models than with any other clustering method. It is not a problem at all unless one wants to be fully Bayesian as

standard MCMC involves the labeling.

One particular class of model based clustering is Latent Class (LC) clustering. Much of the work on LC clustering is based on continuous variables. Generally, these continuous variables are assumed to be Gaussian within latent classes, possibly after applying an appropriate non-linear transformation, see Lazarsfeld and Henry (1968). We are sometimes confronted with other types of indicators like ordinal variables, see Which? example of Section 1.1.1. LC cluster models for ordinal variables assuming (restricted) multinomial distributions for the items are equivalent to standard exploratory LC models for Poisson counts, see Goodman (1974), Böckenholt (2008) and Wedel et al. (1999). Using the general structure of the LC model, it is straightforward to specify cluster models for sets of indicators of different scale types, see Everitt. (1993).

Item Response Theory (IRT), commonly used in psychometrics, provides another framework for ordinal data analysis. IRT provides a framework for evaluating how well assessments work, and how well individual items on assessments work. The most common application of IRT is in education, where psychometricians use it to achieve tasks such as developing and refining exams, and accounting for the difficulties of successive versions of exams, see Hambleton et al. (1991). IRT models are often referred to as latent trait models, developed in the field of sociology, as the latter are virtually identical to IRT models. The term latent is used to emphasize that discrete item responses are taken to be observable manifestations of hypothesized traits, constructs, or attributes, not directly observed, but which must be inferred from the manifest responses. Using the Which? example of Section 1.1.1 we could use IRT, for instance, to incorporate the difficulty of a brand being assigned a higher preference, or a 5 on a 1-5 preference scale, on a given attribute question.

Thus far we have considered model based clustering using mixtures in a classical framework, but in the next chapter we consider mixtures both finite, and infinite, in a Bayesian nonparametric context where the underlying distribution is latent.

## 2.6 Summary

The amount of literature on both frequentist and Bayesian approaches to the multiple comparison problem is vast. Few statistical principles have been as controversial among researchers or practitioners, see O'Neill and Wetherill (1971), O'Brien (1983), and Rothman (1990). But neither approach completely resolves the problem. In essence, frequentist approaches condition on the null hypothesis being true. Therefore, under the conventional $\alpha = 0.05$, it is more difficult to reject the null

hypothesis in favour of the alternative. With the Bayesian approach the prior distribution on the parameters of interest is usually dependent on the circumstances in any particular problem. Assessing the prior distribution that adequately reflects an experimenter's state of knowledge is difficult, more so for a larger number of parameters. However, even in the eyes of the frequentist, the Bayesian position is strong when the prior is specific and reliable. Hence in real applications, researchers should try and quantify their available information of various parameters into a prior distribution. When the prior is not fully specified one can consider using an empirical-Bayesian approach, see Shaffer (1999).

Ultimately, whether multiple comparisons is a problem in a given experiment is purely in the hands of the experimenter and depends on how great the losses are in making wrong decisions. The debate continues.

Cluster analysis techniques are potentially very useful for the exploration of complex multivariate data. The use of this technique requires considerable care if misleading solutions are to be avoided, and much attention needs to be given to the evaluation and validation of results. Given the huge variety of clustering algorithms it is critital we define our research objectives before proceeding in selecting an algorithm that meets our requirements. In the next Chapter we introduce model based clustering using Nonparametric Bayesian modelling. Here we assume that there are an infinite number of latent clusters, some which will be observed in the data. Extensive performance comparisons are then made with this model against K-means, one of the most popular non-model based clustering algorithm, and other MCMs adapted for clustering, see Section 2.4.

# Chapter 3

# Bayesian Nonparametric Methods for Clustering

## 3.1 Introduction

In this chapter we review the current state of nonparametric Bayesian inference. The discussion follows a list of important statistical inference problems from regression to hierarchical models. The discussion is not exhaustive, but the focus will remain on the Dirichet Process (DP) models and an adaptation of the Dirichlet Process Mixture (DPM) which we will use in subsequent chapters as a proposal for model based clustering. We also address implementation issues using various sampling schemes and propose some possible solutions.

## 3.2 Bayesian Nonparametrics

In statistical analysis, the term *nonparametric* usually means to be free of potentially unrealistic and restrictive constraints that are implied by parametric models considered thus far. However, when we incorporate both parametric and nonparametric components into a single model then we have a *semiparametric* model. For example, in linear regression the distribution of the error term is allowed to be arbitrary subject to having a median of zero. There has been an explosion in the number of papers that have been published in this area. In general classical methods make use of permutation and ranking, but more recently, with increasing computation power, jackknifing and resampling methods have played a major role. Both Bayesian and frequentist nonparametric regression modelling, density estimation and smoothing continue to be active areas of research.

With parametric modelling the data are modelled based on a family of probability measures $\{F_\theta : \theta \in \Theta\}$, with their corresponding probability density functions (pdf), say $p(\cdot|\theta)$, where $\Theta$ is finite dimensional. For Bayesian inference we construct a prior for $\theta$ independently of the data. Combining both the likelihood and prior beliefs on $\theta$, we obtain the posterior pdf for $\theta$. Then, based on this posterior, we obtain various characteristics such as posterior means (or medians), standard deviations and probability intervals. If needed, prediction is made for a future observation given the data by integrating out $\theta$ from the product of the posterior and the pdf of a future observation given the data and parameter.

With nonparametric modelling we might assume, for example, that the data are sampled from a completely unknown distribution, $F$, and the goal is to make inferences about functions, or even the pdf, of $F$. We could think of $F$ as belonging to the class of all continuous distributions on the real line for example. In hierarchical modelling $F$ may appear at a higher level in the hierarchy. In contrast Bayesian nonparametric (BNP) inference traditionally refers to Bayesian methods that result in inference comparable to classical nonparametric inference. Such flexible inference is typically achieved by models with many parameters. In fact, a commonly used technical definition of nonparametric Bayesian models are probability models with infinitely many parameters, see Bernardo and Smith (1994).

It was noted by Müller and Quintana (2004) that BNP models can also be used to *robustify* parametric models and to perform sensitivity analysis. For instance in nonparametric regression we can include standard parametric linear regression as a special case. Bayesian modelling accounts for this by constructing a prior that is centred on a parametric regression function. In the same vein, Kleinman and Ibrahim (1978) embedded the family of zero-mean normal models in a broader class of models for random effects in a generalized linear mixed models framework. Also, Berger and Gugliemi (2001) developed general BNP methodology for embedding a family of parametric models in a broader class for determining the adequacy of parametric models.

Our attention now turns to the problem of determining a suitable probability measure to be defined on the class of all distributions on the real line. Possible proposals include the the Dirichlet Process (DP), see Ferguson (1973), the Mixture of DPs (MDP), Antoniak (1974), and the Dirichlet Process Mixture (DPM), Escobar and West (1994). A generalization of the DP is the Pólya Tree (PT), Lavine (1994), which can be extended to Mixtures of PTs (MPT), Lavine (1992), and the Gamma Process, Kalbfeisch (1978), used in the area of survival analysis for modelling the cumulative hazard function in the context of the proportional hazards model, Cox

(1972). The DPM model can also be thought of as a subset of the product partition model, see Quintana and Iglesias (2003). In the next section we look at applications of BNP to clustering. In particular we focus on how the DP can be used for clustering using the DPM model and follow this with a general discussion on implementation issues using MCMC schemes.

## 3.3 Infinite cluster model

We now show how to apply standard hierarchical Bayesian modelling, see Lindley and Smith (1972). Suppose we denote the the data vector for object $j$ with $t$ random samples as $\underline{X}_j = (X_{j1}, \ldots, X_{jt})$, and assume that the data can be characterised by independent samples from some distribution $F(\cdot|\mu_j)$. We can write this model as a two-level hierarchical model

$$\begin{aligned} X_{ji}|\mu_j &\sim F(\cdot|\mu_j) \\ \mu_j|G &\sim G(\cdot), \end{aligned} \tag{3.1}$$

where $X_{ji}$, herein, is conditionally independent given $\mu_j$ and $1 \leq i \leq t$, $1 \leq j \leq m$. In order to carry out Bayesian inference, we need to define a prior distribution $G(\cdot)$ on $\mu_j$ so that statistical inference can be made from this model by finding the posterior $p(\underline{\mu}, G|\underline{X})$, where $\underline{X} = (\underline{X}_1, \ldots, \underline{X}_m)$. There are generally two different ways in which we could specify the distribution $G(\cdot)$. One would be to specify a tractable distribution, such as a Gaussian. The other is to specify $G(\cdot)$ as a weighted collection of $L$ point masses

$$G(\cdot|\underline{w}, \underline{\phi}) = \sum_{k=1}^{L} w_k \delta(\cdot, \phi_k), \tag{3.2}$$

where $\underline{\phi} = (\phi_1, \ldots, \phi_L)$, $\sum_{k=1}^{L} w_k = 1$ and

$$\delta(\mu, \phi) = \begin{cases} 1 & ; \mu = \phi \\ 0 & ; \text{o.w} \end{cases} \tag{3.3}$$

denotes a point mass distribution located at $\phi$. That is, $\underline{\phi}$ refers to the location of the $L$ spikes that make up the distribution $G(\cdot|\underline{w}, \underline{\phi})$.

However, such a model is rather restrictive in that it assumes that there is a fixed

number of clusters. No allowance is made for the idea that, should more objects be observed, more clusters could also be observed. Alternatively, we can start with the assumption that there are an infinite number of latent clusters, some of which are observed in any finite sample. Therefore, to build the infinite cluster model we assume that the objects are drawn from an infinite number of clusters and adapt model (3.2) to

$$G(\cdot|\underline{w}, \underline{\phi}) = \sum_{k=1}^{\infty} w_k \delta(\cdot, \phi_k). \tag{3.4}$$

Although, the number of clusters is unbounded, any finite set of objects will contain representatives from a finite subset of these clusters[1]. In order to apply Bayesian inference in the hierarchical model defined by (3.1) and (3.4), we need to define a prior over the infinite dimensional parameter $(\underline{w}, \underline{\phi})$, where $\underline{w} = (w_1, w_2, \ldots)$ and $w_k$ denotes the $k$th point mass and $\phi_k$ denotes the location of that point mass.

## 3.4 The Dirichlet Process

The foundation for the DP was first provided by Ferguson (1973) and Antoniak (1974). The DP has been widely used as a prior for an unknown distribution in model specification. It takes its name from the fact that it is an infinite dimensional Dirichlet distribution. Recent applications include volatility modelling in finance, Griffin and Steel (2006), and survival analysis, Doss and Huffer (2003).

In Section 2.4 we considered models for clustering under the normal parametric family. However, as we have seen, the goal is to learn from the data without making many assumptions about the distribution that generated them. In a Bayesian setup, this means that we need to set a prior distribution whose support is an infinite dimensional space of probability distributions. The DP has this property, but the sampled distributions are discrete with probability one. We assume that the data are generated from some unknown distribution $G$, in some infinite-dimensional function space. This requires the definition of probability measures on a collection of distribution functions. Such measures are usually referred to as Random Probability Measures (RPMs). One of the most common RPMs is the DP. If $G$ is generated by a DP, then for any partition $A_1, \ldots, A_K$ of the sample space, the vector of random

[1]This model is ideal for our brand clustering example in Section 1.1: brands can vary in a number of ways on a given attribute question, some of which will be observed in a finite sample. With infinitely many clusters, there is always the possibility that a new brand can display behaviour that has never been seen before.

probabilities $P(A_j)$, follows a Dirichlet Distribution (DD)

$$(P(A_1), \ldots, P(A_K)) \sim DD(\alpha G_0(A_1), \ldots, \alpha G_0(A_K)),$$

where $\alpha > 0$ is a measure of dispersion and $G_0$ is a base measure. The DD is defined over the $K-1$ dimensional probability simplex. A $K$-dimensional random vector $\underline{p}$ follows a DD if it has probability density function

$$p(\underline{p}|\underline{\xi}) = \frac{\Gamma(\sum_{j'=1}^{K} \xi_{j'})}{\prod_{j'=1}^{K} \Gamma(\xi_{j'})} \prod_{j=1}^{K} p_j^{\xi_j - 1}, \tag{3.5}$$

where $p_j > 0$, $\xi_j \geq 0$ and $\sum_{j=1}^{k} p_j = 1$. Note that the DD is the conjugate prior for the multinomial distribution. When $K = 2$ we have the Beta distribution. To visualise how random samples from a DD look like, we took samples from a DD when $K = 3$ for which the region is a 2D simplex or triangle, see Figure 3.1. A Dirichlet Process (DP) can be thought of as an 'infinitely decimated' DD. We denote this by $G \sim DP(G_0, \alpha)$. The base measure $G_0$ defines the expectation $\mathbb{E}(G) = G_0$. One attractive property of the DP is its simplicity of posterior updating. Suppose that

$$\mu_1, \ldots, \mu_m | G \sim G,$$

and $G \sim DP(G_0, \alpha)$. Then the posterior distribution of $G$ takes the form

$$G|\mu_1, \ldots, \mu_m \sim DP(G_1, \alpha + m),$$

where

$$G_1(\cdot) = \frac{\alpha G_0(\cdot) + \sum_{j=1}^{m} \delta(\cdot, \mu_j)}{\alpha + m}.$$

The above property makes the DP an attractive proposal in Bayesian hierarchical models too, as we shall see in the next section through the application of DPM using MCMC schemes. A thorough treatment of the DP is given in Ghosal et al. (1999).

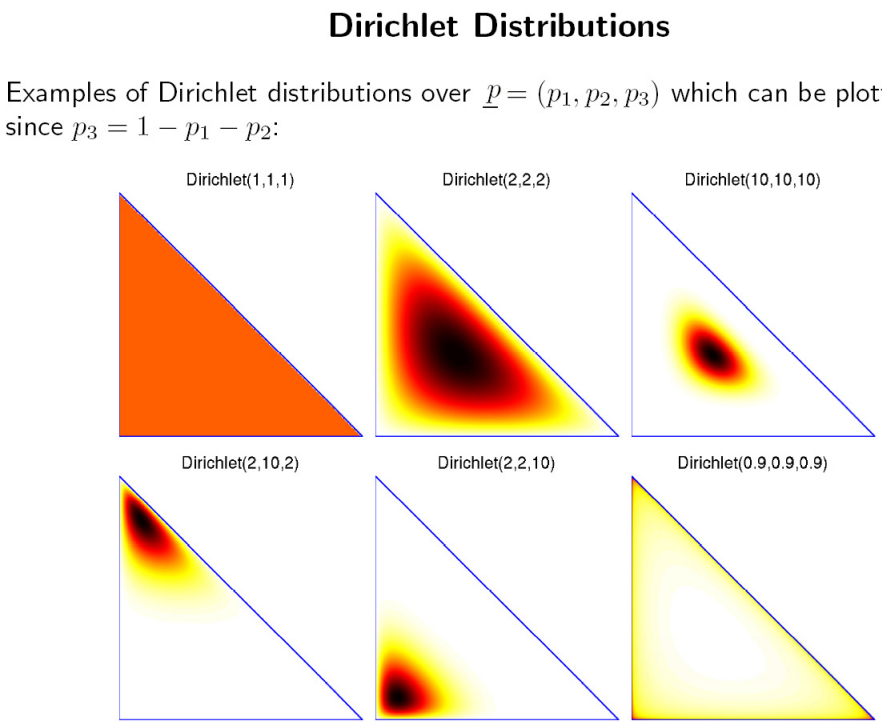


Figure 3.1: Dirichlet Distributions when $K = 3$. **top left**: weight spread uniformly, with $\mathbb{E}[\underline{p}]$=(1/3,1/3,1/3) and $\mathbb{V}[\underline{p}]$=(1/18,1/18,1/18) **top middle**: higher precision of equal weighting across all dimensions, with $\mathbb{E}[\underline{p}]$=(1/3,1/3,1/3) and $\mathbb{V}[\underline{p}]$=(2/63,2/63,2/63) **top right**: even higher precision of equal weighting across all dimensions, with $\mathbb{E}[\underline{p}]$=(1/3,1/3,1/3) and $\mathbb{V}[\underline{p}]$=(2/279,2/279,2/279) **bottom left**: weight more from the middle, with $\mathbb{E}[\underline{p}]$=(1/7,5/7,1/7) and $\mathbb{V}[\underline{p}]$=(2/245,2/147,2/245) **bottom middle**: weight more from the top, with $\mathbb{E}[\underline{p}]$=(1/7,1/7,5/7) and $\mathbb{V}[\underline{p}]$=(2/245,2/245,2/147) **bottom right**: weight mixed from top, middle and bottom, with $\mathbb{E}[\underline{p}]$=(1/3,1/3,1/3) and $\mathbb{V}[\underline{p}]$=(20/333,20/333,20/333). **Note**: Darker shade implies higher weight in that region.

## 3.5 Review of MCMC schemes

Applications of DP hierarchical models are now standard in semiparametric inference. Extending our initial model in (3.1) with a DP prior on $G$ gives

$$\begin{aligned} X_{ji}|\mu_j &\sim F(\cdot|\mu_j) \\ \mu_j|G &\sim G(\cdot) \\ G &\sim DP(G_0, \alpha). \end{aligned} \tag{3.6}$$

Model (3.6) is also known as the DPM. Again, a DP provides a means of placing a distribution on the space of all possible distribution functions.

Inference for DPMs is feasible using MCMC algorithms, in particular using Gibbs sampling techniques, see Ishwaran and James (2001) and Liu (1996). Suppose initially that $G_0$ and $\alpha$ are known. Sampling from $G(\cdot)$ is rather complicated, as shown in Ferguson (1973), which provided the foundation for the DP. There are two alternative characterisations of the DP. The first characterisation is that described in Section 3.4. The second is the *stick-breaking construction*, see Sethuraman (1994). Since his formulation can be thought of as an infinite set of points, $\phi_k$, with corresponding weights, $w_k$, as in (3.4), we specify two separate priors. The stick-breaking process can be illustrated in the following way. First, imagine a stick of length 1, then break it into two pieces and throw away one of those pieces. Continue this process for an infinite number of breaks. We then have an infinite set of stick-lengths that sum to 1 with probability 1. More formally, we assume that at each step of the process the proportion, $u_k$, of the stick broken off follows

$$u_k|\alpha \sim Beta(1, \alpha),$$

where the length of the $k$th stick fragment is

$$w_k = u_k \prod_{l=1}^{k-1}(1 - u_l) \quad k = 2, \ldots, \tag{3.7}$$

where $w_1 = u_1$. A key property of Sethuraman's construction is that it allows us to draw approximate samples from the DP by sampling $\underline{w} = (w_1, w_2, \ldots)$ from the stick-breaking process until $\sum_{h=1}^{L} w_h > 1 - \epsilon$, where $L$ is the number of samples needed until the missing probability mass is less than $\epsilon$. We sample the correspond-

ing $\phi_k$ independently from $G_0$ and treat $(\phi_k, w_k)$, $k = 1, \ldots, L$, as a realisation of the random distribution $G(\cdot)$ given by (3.4). When this construction is used as a computation scheme for the DPM it is known as the conditional method. Figure 3.2 shows distributions sampled from a DP with a standard normal for $G_0$ under three different values of $\alpha$. It is clear from Figure 3.2 that smaller values of $\alpha$ tend to concentrate $G$ on fewer values of $\underline{\phi}$. More specifically when $\alpha$ is very small, $G(\cdot)$ concentrates its mass at one point. Conversely, when $\alpha$ is large $G(\cdot)$ is closer to $G_0$.

To avoid posterior computation for the infinitely-many parameters in (3.7), we can approximate (3.4) by setting $u_L = 1$ from (3.7) and discarding the $L+1, \ldots, \infty$ terms. Other approaches for truncation have been proposed in Ishwaran and Zarepour (2000). These algorithms typically rely on a truncation approximation to the definition of $G$ in (3.4). For a formal justification see Ishwaran and James (2001). Although this approximation can be shown to be highly accurate for DPM models for $L$ sufficiently large, we should be conservative in choosing $L$ to avoid unnecessary computation. Papaspiliopoulos and Roberts (2008) use retrospective sampling to avoid this approximation, see Chapter 7. In the next chapter we propose an alternative scheme which partitions the 'active' and 'non-active' components in $G$ to help address the truncation issue.

An alternative computational scheme for the DPM is the marginal method, which leads to the Pólya urn scheme described by Blackwell and MacQueen (1973), also known as the Chinese Restaurant Process (CRP), see Blackwell and MacQueen (1973). The clustering property of the DP and sample allocation (3.6) can be seen clearly under this representation. In the CRP metaphor, there exists a Chinese restaurant with an infinite numbers of tables. So we start with customer 1 who enters the restaurant and sits at a new table and orders a new dish, $\mu_1$, sampled from $G_0$. Notice that each dish is unique to each table, so the dish can be thought of as the table label. Then the second customer enters and sits at the table occupied by customer 1 with probability $1/(1+\alpha)$ and has the same dish $\mu_1$ or sits at a new table with probability $\alpha/(1+\alpha)$ and orders a new dish $\mu_2$. Therefore the sampled value for $\mu_2$ is

$$\mu_2|\mu_1 \sim \frac{\alpha}{1+\alpha} G_0 + \frac{1}{1+\alpha}\delta(\cdot, \mu_1),$$

where $\delta(\cdot, \mu_1)$ is as defined in (3.3). We carry this process on till the $m$th customer enters, and sits at one of the previously $m-1$ occupied tables with probability $1/(m+\alpha-1)\sum_{j=1}^{m-1}\delta(\cdot, \mu_j)$ or sits at a new table with probability $\alpha/(m+\alpha-1)$

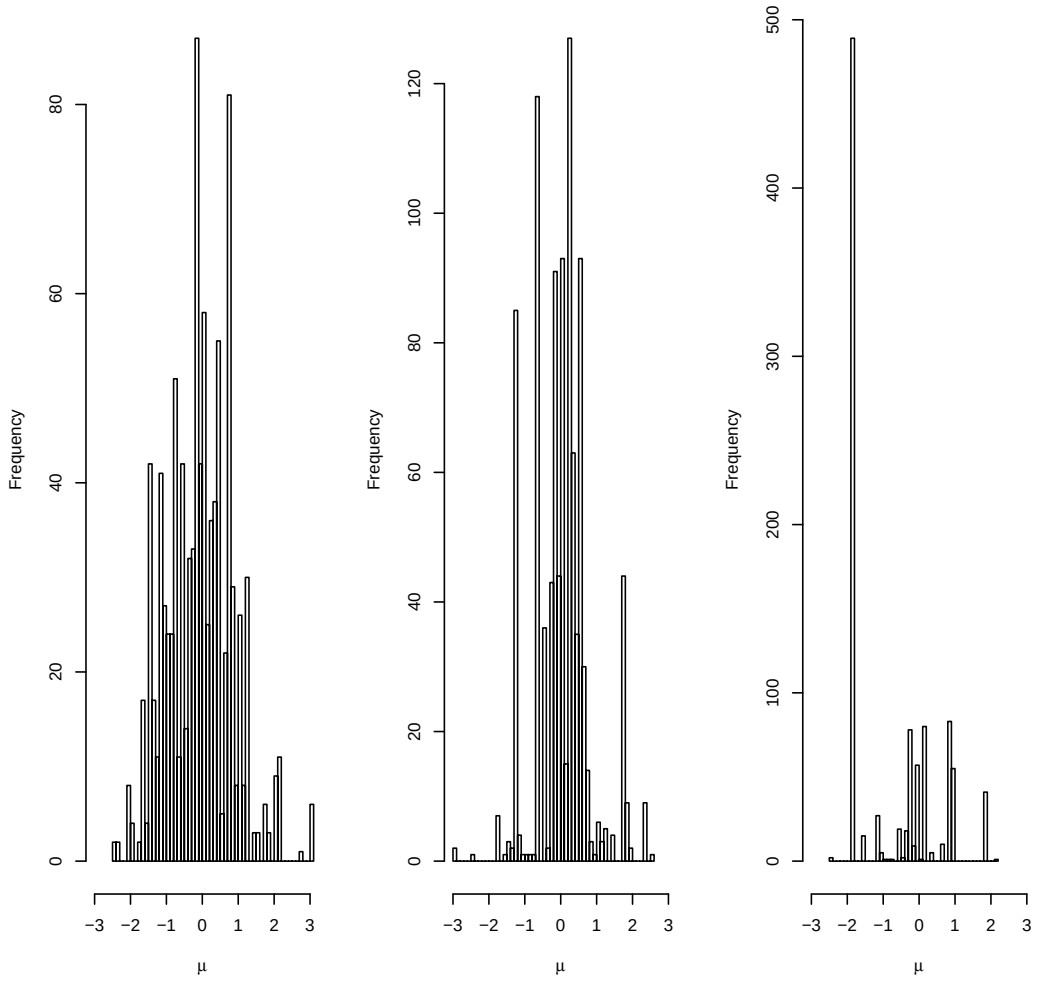


Figure 3.2: Distributions sampled from a DP with a standard normal as the base distribution $G_0(\cdot)$, with dispersion parameter $\alpha = 100$ (left), $\alpha = 20$ (middle), and $\alpha = 5$ (right).

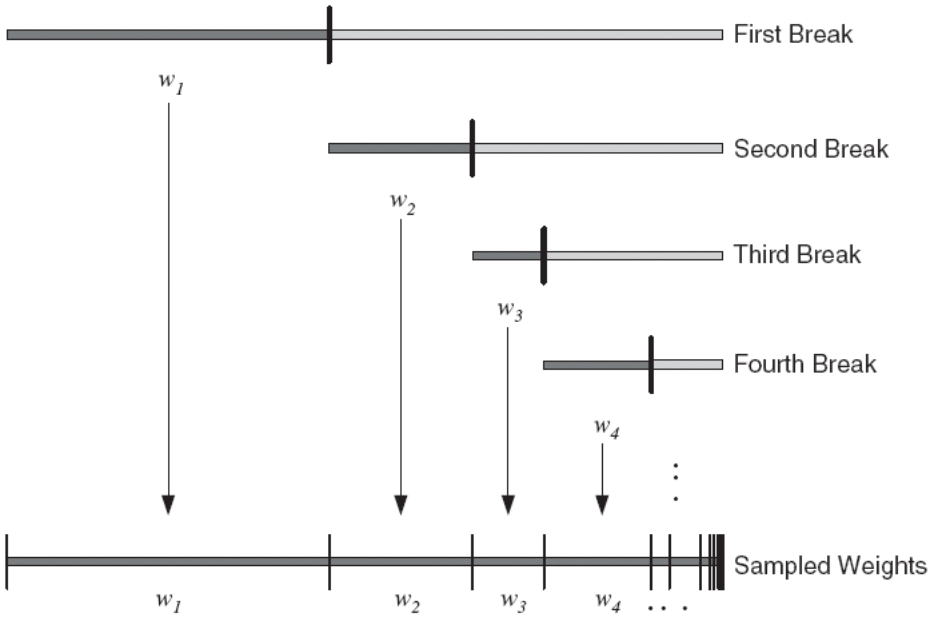


Figure 3.3: A graphical depiction of the stick-breaking process, showing successive breaks of a stick with starting length one, and how the lengths of the pieces correspond to sampled weights.

and orders a new dish. Putting everything together we have

$$\mu_m|\mu_1,\ldots,\mu_{m-1},\alpha,G_0 \sim \frac{1}{m+\alpha-1}\sum_{j=1}^{m-1}\delta(\cdot,\mu_j)+\frac{\alpha}{m+\alpha-1}G_0(\cdot). \tag{3.8}$$

From (3.8) it is clear that customer $m$ would have a higher probability of sitting at a table that already has more customers seated. This clearly illustrates the clustering property of the DP, where new objects are more likely to be placed in clusters that have already been allocated than in a new cluster. We can arrive at (3.8) by integrating out $G$ from (3.6). Extending the number of clusters with the arrival of new data is a desirable property of the CRP. It is made explicit using the CRP metaphor where a new customer can start a new cluster by picking an unoccupied table. This flexibility allows the DPM to achieve model selection automatically. The CRP was generalized to the generalized Pólya urn by West et al. (1994) which is one of the most widely used algorithms. Ishwaran and James (2001) extended this approach to a general class of stick-breaking measures.

One of the criticisms of the conditional method is that it is an inconvenient formulation for computational purposes, since it requires a large number of $\phi_k$ and

$u_k$ values to be maintained. However it has two considerable advantages over the marginal method. First, it does not rely on being able to integrate out analytical components, such as $G$, in the hierarchical model and, therefore, it is more flexible for current and future enhancements of the basic model. Such extensions include more general stick-breaking random measures, and modelling dependence of the data on covariates, see Dunson and Park (2008). Also note that, due to the sequential conditional updating of $\mu_j$ in the marginal method, we introduce dependencies between the $\mu_j$, which will increase the convergence time of the MCMC sampling scheme.

The stick-breaking representation is probably the most versatile definition of the DP. It has been exploited to generate efficient alternative samplers like the Blocked Gibbs sampler, see Ishwaran and James (2001), which relies on a finite-sum approximation, and the Retrospective sampler of Papaspiliopoulos and Roberts (2008), which does not require truncation. It is also the starting point for the definition of many generalizations that allow dependence across a collection of distributions, including the Dependent Dirichlet Process (DDP), see MacEachern (2000), and $\pi$DDP, see Griffin and Steel (2006).

## 3.6 Other Random Processes

There are other extensions to the standard stick-breaking construction in (3.7) which include sampling $u_k|\alpha, \beta \sim Beta(\alpha, \beta)$, the so called Beta two-parameter process in Ishwaran and Zarepour (2000). Other extensions include the Pitman and Yor (1997) process.

As we saw briefly in Section 3.2 there is a wide class of random processes that can be defined as an alternative to the DP. In particular, two generalizations of the DP are the PT and the Gamma Process (GP). The GP is a continuous time stochastic process that starts at $X_0 = 0$ and has independent Gamma increments. The GP can be generalized to the generalized GP, introduced by Brix (1999), for constructing shot noise Cox Process. A generalized GP $GG(\beta, \sigma)$ depends on two parameters $\sigma \in (0, 1)$ and $\beta > 0$. For a more formal specification of $\sigma$ and $\beta$ see Lijoi et al. (2007). For a given $\sigma$ and $\beta$, Lijoi et al. (2007) showed that the generalized GP induces a partition and provides the distribution of the number of distinct clusters $K_m$. In Korwar and Hollander (1973) it was shown that the number $K_m$ of clusters

that are induced by the DP is governed by

$$\frac{K_m}{\log(m)} \to \alpha,$$

where $\alpha > 0$ is the dispersion parameter as before. The influence of $\beta$ and $\sigma$ on $K_m$ was investigated by Lijoi et al. (2007). They showed that the bigger $\beta$ is the larger the expected number of clusters tends to be. In contrast, $\sigma$ controls the flatness of the distribution of $K_m$. So the larger $\sigma$ is the flatter the distribution of $K_m$, suggesting that large values of $\sigma$ yields a non-informative prior for $K_m$. Lijoi et al. (2007) also propose a reasonable strategy for the prior specification of $(\beta, \sigma)$ would be to fix $\mathbb{E}_{\beta,\sigma}[K_m]$ equal to the prior opinion on the expected number of clusters. In Chapter 6 we carry out a more detailed review and extension of the choice of prior for the expected number of clusters in the DP.

## 3.7 Summary

In this chapter we have reviewed some important aspects of nonparametric Bayesian inference, with the focus on understanding the DP and how it can be incorporated into a DPM framework for clustering purposes. As we have observed, there are some methodological challenges here. In particular we see that one of the difficulties of implementing the conditional method is the truncation that is required of the infinite dimensional vectors $\underline{u}$ and $\underline{\phi}$. Although there have been some authors who have addressed this problem, we propose a similar approach to that used by Papaspiliopoulos and Roberts (2008) where they consider the *active* and *non-active* components of $\underline{u}$ and $\underline{\phi}$ separately using retrospective sampling, see Chapter 7. In the next chapter we consider how the DPM can be used when we have Normal data.

# Chapter 4

# Dirichlet Process Mixture for Normal Data

## 4.1 Introduction

In this chapter we focus on applying the DPM, introduced in the previous chapter, where the distribution of the data is taken to be Normal. We implement this model using the conditional method and extend the framework to address the problems encountered with truncation as we saw in Section 3.5. We conclude with two simulation studies. One study compares our DPM model against an alternative GP model used in the simulation by Lijoi et al. (2007). In the other study we make comparisons of our DPM model against all the other proposals introduced in Section 2.4. To allow detailed comparisons between methods we assess each method on three different measures. Finally, we compare the performance of the two most popular approaches for sampling from a DP, namely the conditional and marginal schemes, as seen in Section 3.5.

## 4.2 Dirichlet Process for Normal Data

Assume we have objects, each with some random observations, with corresponding means $\mu_j$ drawn from an infinite number of clusters, where we take a weighted combination of an infinite number of point masses, $w_k$, on points $\phi_k$ so that

$$P(\mu_j = \mu | \underline{w}, \underline{\phi}) = \sum_{k=1}^{\infty} w_k \delta(\mu, \phi_k), \tag{4.1}$$

where $\delta(\mu, \phi)$ is defined by (3.3). An advantage of an infinite cluster model over a finite model is that a new object can be assigned to a new cluster, therefore allowing the objects to vary in a number of ways, some of which will be observed from the data. Any finite set of objects will contain representatives from a finite number of these clusters. In this chapter we choose the weights $w_k$ corresponding to a DP prior for $G$. We define the relevant priors for $\phi_k$ and $w_k$.

Herein we denote $G|G_0, \alpha \sim DP(G_0, \alpha)$, where $G_0$ represents our belief about the kind of values that best represent the behaviour of $\mu_j$. The full data model and priors for all the parameters in our model are as follows:

$$\begin{aligned}
X_{ji}|\sigma^2, \mu_j &\sim N\left(\mu_j, \sigma^2\right) \\
\sigma^2|v_0, \sigma_0^2 &\sim InvGamma\left(\frac{v_0}{2}, \frac{v_0\sigma_0^2}{2}\right) \\
\mu_j|G &\sim G(\cdot) \\
G|G_0, \alpha &\sim DP(G_0, \alpha) \\
\alpha|a, b &\sim Gamma(a, b) \\
G_0|\mu_0, k_0 &\equiv N\left(\mu_0, \frac{1}{k_0}\right) \\
\mu_0|k_0 &\sim N\left(\mu_1^*, \frac{\sigma_2^2}{k_0}\right) \\
k_0 &\sim Gamma\left(\frac{v_1}{2}, \frac{v_1\sigma_1^2}{2}\right).
\end{aligned} \tag{4.2}$$

The common variance of $X_{ji}$, $\sigma^2$, is assigned a prior that is conjugate to the normal, i.e. an inverse gamma. Sampling the $\mu_j$ from a realization $G$ of a DP induces clustering, as explored in Section 3.5. The level of clustering is controlled by the dispersion parameter $\alpha$, which is also known as the smoothing parameter. A common choice for the $\alpha$ prior is a Gamma distribution. Specifying a prior on $\alpha$ allows us to learn the number of clusters from the data as well capturing our prior beliefs about the number of clusters. We explore the prior specification of $\alpha$ in more detail in Chapter 6. The location parameter $\mu_0$ and precision $k_0$ of $G_0$ are themselves given priors that are conjugate to $G_0$.

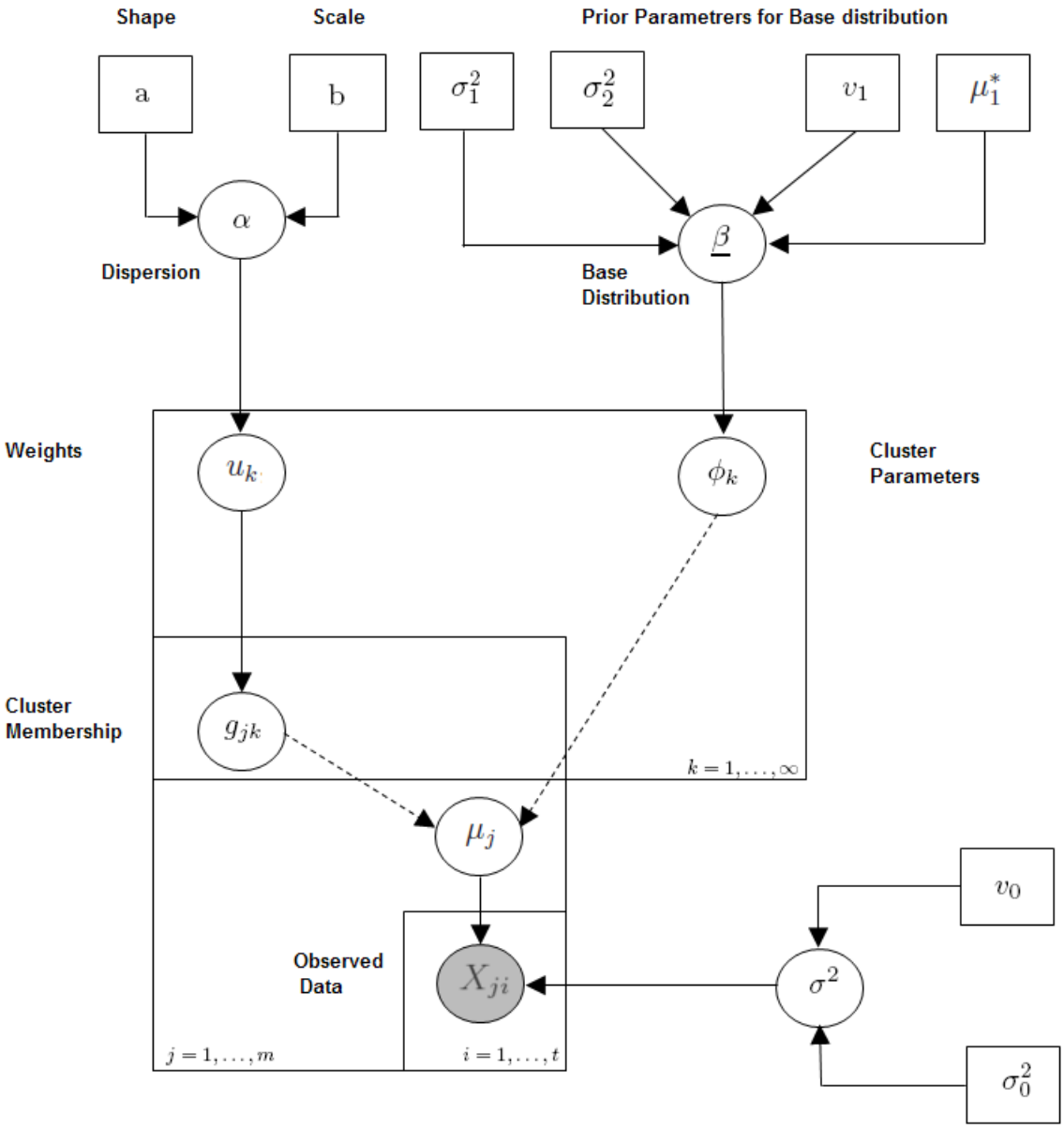


Figure 4.1: Dependencies in the infinite cluster normal model. Circles are random variables, squares denote known parameter values, and plates indicate a set of independent replicates of the random variables shown inside them. Dashed lines indicate the child node is derived from its parent nodes.

## 4.3 Gibbs Sampling

### 4.3.1 Conditional Method

The model in (4.2), which we refer to as the Dirichlet Process Normal Mixture (DPNM) model, can be fitted using Gibbs sampling, which is a Markov chain Monte Carlo method of sampling from the posterior distribution that uses the full distributions, conditional on all other variables in the Bayesian model. See, for example Neal (2000) or Gilks et al. (1995). The idea of Gibbs sampling is to fix all variables in the posterior except one variable, or group of variables, and sample that variable, or group of variables, from its conditional posterior distribution. Repeat this for the other variables, each time treating one variable as random and conditioning on the most recently updated values for the others. Then it can be shown that for a large enough run of this chain a random sample from the joint posterior distribution is generated.

To achieve this we use the stick-breaking representation of the DP, described in Section 3.5. Thus, given the set of parameters $\{\underline{\beta}, \sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}\}$, where $\underline{\beta} = (\mu_0, k_0)$, $\underline{\phi} = (\phi_1, \ldots)$ and $\underline{u} = (u_1, \ldots)$, see (3.7). Also let $(\underline{g} = g_{jk}, j = 1, \ldots, m, k = 1, 2, \ldots)$, and $g_{jk}$ is the cluster indicator variable

$$g_{jk} = \begin{cases} 1 & ; \text{If the } j\text{th object is in the } k\text{th cluster} \\ 0 & ; \text{o.w.} \end{cases} \tag{4.3}$$

Under model (4.2), the DP provides a prior for the distribution of $\mu_j$. A graphical model for (4.1) is illustrated in Figure 4.1. Herein, under a graphical model, the joint probabilities of the random model parameters factor into a product of conditional distributions. Therefore, any two nodes are conditionally independent given the values of their parents. Since $G_0 \equiv G_0(\underline{\beta})$ and using relationships from Figure 4.1, the joint posterior density can be written as

$$p(\underline{\beta}, \sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}|\underline{X}) \propto p(\underline{\beta})p(\sigma^2)p(\alpha)p(\underline{u}|\alpha)p(\underline{\phi}|\underline{\beta})p(\underline{g}|\underline{u})p(\underline{X}|\underline{g}, \underline{\phi}, \sigma^2).$$

Then it follows that

$$\begin{aligned}
p(\underline{\beta}|-) &\propto p(\underline{\beta})p(\underline{\phi}|\underline{\beta}) \\
p(\sigma^2|-) &\propto p(\sigma^2)p(\underline{X}|\underline{g},\underline{\phi},\sigma^2) \\
p(\alpha|-) &\propto p(\alpha)p(\underline{u}|\alpha) \\
p(\underline{\phi}|-) &\propto p(\underline{\phi}|\underline{\beta})p(\underline{X}|\underline{g},\underline{\phi},\sigma^2) \\
p(\underline{u}|-) &\propto p(\underline{u}|\alpha)p(\underline{g}|\underline{u}) \\
p(\underline{g}|-) &\propto p(\underline{g}|\underline{u})p(\underline{X}|\underline{g},\underline{\phi},\sigma^2),
\end{aligned} \tag{4.4}$$

where $-$ denotes the full conditionals on the other variables, otherwise we use a block sampler as we shall see later. For notational convenience we use the same symbols for both the random variables as well as their values from now on. We now compute the various components in (4.4). First, the likelihood is

$$\begin{aligned}
p(\underline{X}|\underline{g},\underline{\phi},\sigma^2) &\propto \prod_{j=1}^{m}\left(\sigma^2\right)^{-\frac{t}{2}}\exp\left[-\frac{1}{2\sigma^2}\left\{\sum_{i=1}^{t}(X_{ji}-\bar{X}_j)^2+t(\bar{X}_j-\mu_j)^2\right\}\right] \\
&= \left(\sigma^2\right)^{-\frac{tm}{2}}\exp\left[-\frac{1}{2\sigma^2}\left\{(t-1)\sum_{j=1}^{m}s_j^2+t\sum_{j=1}^{m}\sum_{k=1}^{L}(\bar{X}_j-\phi_k)^2 g_{jk}\right\}\right].
\end{aligned} \tag{4.5}$$

We are now in a position to find the various conditional posteriors in (4.4). First we find distribution $p(\underline{u}|-)$. But, since $u_k|\alpha \sim Beta(1,\alpha)$, we have

$$p(\underline{u}|\alpha) \propto \prod_{k=1}^{L}\alpha(1-u_k)^{\alpha-1}. \tag{4.6}$$

Also,

$$p(\underline{g}|\underline{u}) \propto \prod_{j=1}^{m}\prod_{k=1}^{L}w_k^{g_{jk}} = \prod_{k=1}^{L}w_k^{\sum_{j=1}^{m}g_{jk}} = \prod_{k=1}^{L}w_k^{r_k} \tag{4.7}$$

where $r_k$ denotes the number of $\mu_j$'s that are in the $k$th cluster, with $\sum_{k=1}^{L}r_k = m$.

It follows from (3.7) that

$$\begin{aligned} p(\underline{u}|-) &\propto \prod_{k=1}^{L}(1-u_k)^{\alpha-1}\left\{u_k\prod_{z=1}^{k-1}(1-u_z)\right\}^{r_k} \\ &= \prod_{k=1}^{L}u_k^{(r_k+1)-1}(1-u_k)^{\left(\alpha+\sum_{z=k+1}^{L}r_z\right)-1}. \end{aligned} \tag{4.8}$$

We see that (4.8) is proportional to the density of the $Beta\left(r_k+1, \alpha+\sum_{z=k+1}^{L}r_z\right)$ distribution.

Next, from (4.5) and (4.6), we obtain

$$\begin{aligned} p(\underline{g}|-) &\propto p(\underline{g}|\underline{u})p(\underline{X}|\underline{g},\underline{\phi},\sigma^2) \\ &= \prod_{k=1}^{L}w_k^{r_k}\left[\prod_{j=1}^{m}\prod_{k'=1}^{L}\exp\left\{-\frac{t}{2\sigma^2}(\bar{X}_j-\phi_{k'})^2 g_{jk'}\right\}\right] \\ &\propto \prod_{j=1}^{m}\prod_{k=1}^{L}\left[w_k\exp\left\{-\frac{t}{2\sigma^2}(\bar{X}_j-\phi_k)^2\right\}\right]^{g_{jk}}. \end{aligned} \tag{4.9}$$

Therefore we obtain

$$p(\underline{g}|-) = \prod_{j=1}^{m}\prod_{k=1}^{L}\tilde{w}_{jk}^{g_{jk}},$$

where

$$\tilde{w}_{jk} = \frac{w_k\exp\left\{-\frac{t}{2\sigma^2}(\bar{X}_j-\phi_k)^2\right\}}{\sum_{k'=1}^{L}w_{k'}\exp\left\{-\frac{t}{2\sigma^2}(\bar{X}_j-\phi_{k'})^2\right\}}.$$

Hence $g_{jk}=1$ when an object mean falls in cluster $k$ with probability $\tilde{w}_{jk}$, where $\sum_{k=1}^{L}\tilde{w}_{kj}=1$.

We next obtain $p(\underline{\phi}|-)$. Since $\phi_k \sim N(\mu_0, k_0^{-1})$, we obtain from (4.4) and (4.5)

$$\begin{aligned} p(\underline{\phi}|-) &\propto \prod_{k=1}^{L}\exp\left[-\frac{1}{2\sigma^2}\left\{k_0\sigma^2\left(\phi_k-\mu_0\right)^2 + t\sum_{j=1}^{m}\left(\bar{X}_j-\phi_k\right)^2 g_{jk}\right\}\right] \\ &\propto \prod_{k=1}^{L}\exp\left[-\frac{k_0\sigma^2+tr_k}{2\sigma^2}\left\{\left(\phi_k-\frac{k_0\sigma^2\mu_0+t\sum_{j=1}^{m}\bar{X}_j g_{jk}}{k_0\sigma^2+tr_k}\right)^2\right\}\right]. \end{aligned}$$

Therefore

$$\phi_k|- \sim N\left(\frac{k_0\sigma^2\mu_0 + t\sum_{j=1}^m \bar{X}_j g_{jk}}{k_0\sigma^2 + tr_k}, \frac{\sigma^2}{k_0\sigma^2 + tr_k}\right). \tag{4.10}$$

Next, writing $p(\underline{\beta}) = p(\mu_0|k_0)p(k_0)$, we update $(\mu_0, k_0)$ in a block. From (4.2) and (4.4) we see that

$$\begin{aligned} p(\underline{\beta}|-) &\propto k_0^{\frac{1}{2}} e^{-\frac{k_0}{2\sigma_2^2}(\mu_0-\mu_1^*)^2} k_0^{\frac{v_1}{2}-1} e^{-\frac{v_1\sigma_1^2}{2}k_0} e^{-\frac{k_0}{2}\left\{\sum_{k=1}^L(\phi_k-\mu_0)^2\right\}} k_0^{\frac{L}{2}} \\ &= k_0^{\frac{v_1+L+1}{2}-1} \exp\left[-\frac{k_0}{2}\left\{\frac{1}{\sigma_2^2}(\mu_0-\mu_1^*)^2 + \sum_{k=1}^L(\phi_k-\mu_0)^2 + v_1\sigma_1^2\right\}\right]. \end{aligned}$$

It follows that

$$\mu_0|- \sim N\left(\frac{\sum_{k=1}^L \phi_k + \frac{\mu_1^*}{\sigma_2^2}}{L + \frac{1}{\sigma_2^2}}, \frac{1}{k_0\left(L + \frac{1}{\sigma_2^2}\right)}\right) \tag{4.11}$$

and, by noting that

$$p(\mu_0|-) = \frac{p(\underline{\beta}|-)}{p(k_0|\sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}, \underline{X})},$$

we see that

$$p(k_0|\sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}, \underline{X}) \sim Gamma\left[\frac{v_1+L}{2}, \frac{1}{2}\left\{\frac{(\mu_1^*)^2}{\sigma_2^2} + \sum_{k=1}^L \phi_k^2 - \frac{\left(\sum_{k=1}^L \phi_k + \frac{\mu_1^*}{\sigma_2^2}\right)^2}{L + \frac{1}{\sigma_2^2}} + v_1\sigma_1^2\right\}\right]. \tag{4.12}$$

Here we see that $p(\underline{\beta}|-)$ is generated in a block where we first generate (4.12) followed by (4.11).

Again, from (4.2), (4.4) and (4.5) it follows after some algebra that

$$\begin{aligned} p(\sigma^2|-) &\propto p(\sigma^2)p(\underline{X}|\underline{g}, \underline{\phi}, \sigma^2) \\ &\propto (\sigma^2)^{-\left(\frac{v_0+tm}{2}+1\right)} \exp\left(-\frac{c^*}{2\sigma^2}\right), \end{aligned} \tag{4.13}$$

where

$$c^* = (t-1)\sum_{j=1}^m s_j^2 + t\sum_{j=1}^m\sum_{k=1}^L (\bar{X}_j - \phi_k)^2 g_{jk} + v_0\sigma_0^2.$$

Therefore

$$\sigma^2|- \sim InvGamma\left(\frac{v_0+tm}{2}, \frac{c^*}{2}\right). \tag{4.14}$$

Finally from (4.2), (4.4) and (4.6)

$$\begin{aligned} p(\alpha|-) &\propto p(\alpha)p(\underline{u}|\alpha) \\ &\propto \alpha^{a+L-1}\exp\left[-\left\{b-\sum_{k=1}^{L}\log(1-u_k)\right\}\alpha\right] \end{aligned}$$

and it follows that

$$\alpha|- \sim Gamma(a+L, b-\sum_{k=1}^{L}\log(1-u_k)). \tag{4.15}$$

Having derived all the posterior conditionals in our model we now use the Gibbs sampler, a special case of the Metropolis-Hastings algorithm, and thus an example of a Markov Chain Monte Carlo algorithm, to sweep through $\left\{\underline{\beta}, \sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}\right\}$ in order for a given iteration. Over time these will be a sample from the full posterior $p(\underline{\beta}, \sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g}|\underline{X})$.

Much of the implementation thus far is an application of the work by Ishwaran and James (2002). Specifically, as we do, they used a block Gibbs sampling strategy along with an approach to approximate $L$. In addition they assumed unequal within-cluster variance by using $\sigma_j^2$ instead of $\sigma^2$ in model (4.2). An alternatively implementation based on the marginal method, as discussed in Section 3.5, is presented in the work by Escobar and West (1995).

We are constrained by the fact that if we are given $m$ objects then it is impossible to observe more than $m$ clusters. In theory $L = \infty$, but very large values of $L$ will cause dependency problems in the posteriors (4.8)-(4.15), thus leading to slower, or in the worst case halting, convergence to the full posterior. In particular consider the conditional posterior for $\alpha$. Then we see that

$$\mathbb{E}[\alpha|-] = \frac{1}{\frac{1}{L}\left(-\sum_{k=1}^{L}\log(1-u_k)\right)} \to \frac{1}{M} \tag{4.16}$$

as $L \to \infty$, where $M = \mathbb{E}[-\log(1-U_k)|\alpha']$, $\alpha'$ is the previous value of $\alpha$, $U \sim B(1, \alpha)$ and

$$\mathbb{V}[\alpha|-] = \frac{1}{\frac{1}{L}\left(-\sum_{k=1}^{L}\log(1-u_k)\right)^2} \sim \frac{1}{LM^2} \to 0. \tag{4.17}$$

Therefore we see that the chain of values for $\alpha$ become constant as $L \to \infty$. This clearly illustrates the link between the $\underline{u} = (u_1, \ldots)$ and $\alpha$ through the *ergodicity*

*constraint*, see Papaspiliopoulos et al. (2007). In the next section we address this issue by proposing an alternative sampler where we split the generated $L$ components into 'active' and 'non-active' parts.

### 4.3.2 A modified Gibbs Sampler

As we saw in the previous section there are dependency issues that arise in some of the conditional posteriors (4.8)-(4.15) when $L$ becomes large. To remedy this, we could try performing block updates for $(\alpha, \underline{u})$ and $(\underline{\beta}, \underline{\phi})$. The full conditionals for $(\alpha, \underline{u})$ and $(\underline{\beta}, \underline{\phi})$ are

$$\begin{aligned} p(\alpha, \underline{u}|-) &\propto p(\alpha)p(\underline{u}|\alpha)p(\underline{g}|\underline{u}) \\ p(\underline{\beta}, \underline{\phi}|-) &\propto p(\underline{\beta})p(\underline{\phi}|\underline{\beta}). \end{aligned} \tag{4.18}$$

We have $p(\alpha, \underline{u}|-) = p(\alpha|\underline{\phi}, \underline{g}, \underline{\beta}, \sigma^2, \underline{X})p(\underline{u}|-)$, where from (4.2) and (4.8) we see that

$$\begin{aligned} p(\alpha|\underline{\phi}, \underline{g}, \underline{\beta}, \sigma^2, \underline{X}) &\propto p(\alpha) \int p(\underline{u}|\alpha)p(\underline{g}|\underline{u})d\underline{u} \\ &= p(\alpha) \int \prod_{k=1}^{L} u_k^{(r_k+1)-1}(1-u_k)^{(\alpha+R_k)-1}d\underline{u} \\ &= p(\alpha) \prod_{k=1}^{L} \frac{\Gamma(r_k+1)\Gamma(\alpha+R_k)}{\Gamma(\alpha+R_{k-1}+1)}, \end{aligned} \tag{4.19}$$

where $R_k = \sum_{z=k+1}^{L} r_z$. Also

$$p(\underline{\beta}, \underline{\phi}|-) = p(\underline{\beta}|\sigma^2, \alpha, \underline{u}, \underline{g}, \underline{X})p(\underline{\phi}|-),$$

where from (4.2) we see that

$$\begin{aligned} p(\underline{\beta}|\sigma^2, \alpha, \underline{u}, \underline{g}, \underline{X}) &\propto p(\underline{\beta}) \int p(\underline{\phi}|\underline{\beta})p(\underline{X}|\underline{g}, \underline{\phi}, \sigma^2)d\underline{\phi} \\ &\propto k_0^{\frac{v_1+1}{2}-1} \exp\left\{-\frac{k_0}{2}\left(\frac{(\mu_0-\mu_1^*)^2}{\sigma_2^2} + v_1\sigma_1^2\right)\right\} \prod_{k=1}^{L}(k_0\sigma^2 + tr_k)^{-\frac{1}{2}}. \end{aligned} \tag{4.20}$$

However, we see that the conditionals in (4.19)-(4.20) are non-standard distributions. Instead, we will partition $\underline{u}$ and $\underline{\phi}$ into 'active' and 'non-active' components, where

$$m^* = \max\left\{k : \sum_{z=1}^{k} r_z = m\right\}$$

are active and $L - m^*$ non-active. We define the active and non-active cases for $\underline{u}$ as $\underline{u}_{(1)} = (u_1, \ldots, u_{m^*})$ and $\underline{u}_{(2)} = (u_{m^*+1}, \ldots, u_L)$ respectively. In the same way, we define $\underline{\phi}_{(1)} = (\phi_1, \ldots, \phi_{m^*})$ and $\underline{\phi}_{(2)} = (\phi_{m^*+1}, \ldots, \phi_L)$[1]. We see that (4.19) yields a standard distribution when we integrate out $\underline{u}_{(2)}$ and similarly when we integrate out $\underline{\phi}_{(2)}$ in (4.20).

From (4.8) and the definition of $m^*$, we see that

$$\underline{u}_{(1)k}|- \sim Beta\left(r_k + 1, \alpha + R_k\right). \tag{4.21}$$

By integrating out $\underline{u}_{(2)}$ we have

$$\begin{aligned} p(\alpha|\underline{u}_{(1)}, \underline{\phi}_{(1)}, \underline{\phi}_{(2)}, \underline{g}, \underline{\beta}, \sigma^2, \underline{X}) &\propto p(\alpha)p(\underline{u}_{(1)}|\alpha)\int p(\underline{u}_{(2)}|\alpha)d\underline{u}_{(2)} \\ &= p(\alpha)p(\underline{u}_{(1)}|\alpha) \\ &\propto \alpha^{a+m^*-1}e^{-b\alpha}\prod_{k=1}^{m^*}(1-u_k)^{\alpha-1} \\ &= \alpha^{a+m^*-1}e^{-\left\{b-\sum_{k=1}^{m^*}\log(1-u_k)\right\}\alpha}, \end{aligned}$$

from (4.8), so that

$$\alpha|\underline{u}_{(1)}, \underline{\phi}_{(1)}, \underline{\phi}_{(2)}, \underline{g}, \underline{\beta}, \sigma^2, \underline{X} \sim Gamma\left(a + m^*, b - \sum_{k=1}^{m^*}\log(1-u_k)\right). \tag{4.22}$$

We are therefore able to generate $(\alpha, \underline{u}_{(2)})$ in a block by first generating $\alpha$ from (4.22) followed by $\underline{u}_{(2)}$ from $Beta(1, \alpha)$, which follows from (4.8) and the definition of $m^*$. Comparing (4.15) and (4.22) we see that $m^* << L$ therefore avoiding the ergodicity constraint as seen in the last section. Next we generate the $m^*$ components of $\underline{\phi}_{(1)}$

[1]It is possible to observe a component that is unoccupied in the active set, but the weights on these are negligible

from

$$N\left(\frac{k_0\sigma^2\mu_0 + t\sum_{j=1}^{m}\bar{X}_j g_{jk}}{k_0\sigma^2 + tr_k}, \frac{\sigma^2}{k_0\sigma^2 + tr_k}\right).$$

By integrating out $\underline{\phi}_{(2)}$ we have

$$\begin{aligned} p(\underline{\beta}|\sigma^2, \underline{\phi}_{(1)}, \alpha, \underline{u}_{(1)}, \underline{u}_{(2)}, \underline{g}, \underline{X}) &\propto p(\underline{\beta})p(\underline{\phi}_{(1)}|\underline{\beta}) \int p(\underline{\phi}_{(2)}|\underline{\beta})d\underline{\phi}_{(2)} \\ &\propto k_0^{\frac{v_1+m^*+1}{2}-1} e^{-\frac{k_0}{2}\left\{\frac{(\mu_0-\mu_1^*)^2}{\sigma_2^2}+\sum_{k=1}^{m^*}(\phi_k-\mu_0)^2+v_1\sigma_1^2\right\}} \end{aligned}$$

from (4.20), from which it follows

$$\mu_0|- \sim N\left(\frac{\frac{\mu_1^*}{\sigma_2^2} + \sum_{k=1}^{m^*}\phi_k}{\frac{1}{\sigma_2^2} + m^*}, \frac{1}{k_0\left(\frac{1}{\sigma_2^2} + m^*\right)}\right).$$

Also, by noting that

$$p(\mu_0|-) = \frac{p(\underline{\beta}|\sigma^2, \underline{\phi}_{(1)}, \alpha, \underline{u}_{(1)}, \underline{u}_{(2)}, \underline{g}, \underline{X})}{p(k_0|\underline{\phi}_{(1)}, \alpha, \underline{u}_{(1)}, \underline{u}_{(2)}, \underline{g}, \underline{X})},$$

we see that

$$k_0|\underline{\phi}_{(1)}, \alpha, \underline{u}_{(1)}, \underline{u}_{(2)}, \underline{g}, \underline{X} \sim Gamma\left[\frac{v_1 + m^*}{2}, \frac{1}{2}\left\{\frac{(\mu_1^*)^2}{\sigma_2^2} + \sum_{k=1}^{m^*}\phi_k^2 - \frac{\left(\frac{\sum_{k=1}^{m^*}\phi_k}{\sigma^2} + \frac{\mu_1^*}{\sigma_2^2}\right)^2}{m^* + \frac{1}{\sigma_2^2}} + v_1\sigma_1^2\right\}\right].$$

We can now generate $(\underline{\beta}, \underline{\phi}_{(2)})$ as a block from

$$k_0|\underline{\phi}_{(1)}, \alpha, \underline{u}_{(1)}, \underline{u}_{(2)}, \underline{g}, \underline{X} \sim Gamma\left[\frac{v_1 + m^*}{2}, \frac{c^*}{2}\right],$$

where

$$c^* = \frac{(\mu_1^*)^2}{\sigma_2^2} + \sum_{k=1}^{m^*}\phi_k^2 - \frac{\left(\frac{\sum_{k=1}^{m^*}\phi_k}{\sigma^2} + \frac{\mu_1^*}{\sigma_2^2}\right)^2}{m^* + \frac{1}{\sigma_2^2}} + v_1\sigma_1^2,$$

$$\mu_0|- \sim N\left(\frac{\frac{\mu_1^*}{\sigma_2^2} + \sum_{k=1}^{m^*}\phi_k}{\frac{1}{\sigma_2^2} + m^*}, \frac{1}{k_0\left(\frac{1}{\sigma_2^2} + m^*\right)}\right),$$

and

$$\underline{\phi}_{(2)}|- \sim N\left(\mu_0, \frac{1}{k_0}\right).$$

Next we generate $\sigma^2$ from

$$\sigma^2|- \sim InvGamma\left(\frac{v_0 + tm}{2}, \frac{(t-1)\sum_{j=1}^{m} s_j^2 + t\sum_{j=1}^{m}\sum_{k=1}^{m^*}(\bar{X}_j - \phi_k)^2 g_{jk} + v_0\sigma_0^2}{2}\right).$$

Finally

$$p(\underline{g}|-) = \prod_{j=1}^{m}\prod_{k=1}^{L} \tilde{w}_{jk}^{g_{jk}}, \tag{4.23}$$

where

$$\tilde{w}_{jk} = \left[\frac{w_k \exp\left\{-\frac{t}{2\sigma^2}\left(\bar{X}_j - \phi_k\right)^2\right\}}{\sum_{k'=1}^{L} w_{k'} \exp\left\{-\frac{t}{2\sigma^2}\left(\bar{X}_j - \phi_{k'}\right)^2\right\}}\right]$$

and we can find the values of the respective $r_k = \sum_{j=1}^{m} g_{jk}$ from the generated $\underline{g}$.

Having formulated all the posterior conditionals in our model, we are in a position to start the sampler by first finding

$$m^{*(l-1)} = \max\left\{k : \sum_{z=1}^{k} r_z^{(l-1)} = m\right\},$$

where $l = 1, \ldots, T$ is the number of after burn-in iterations of the Gibbs sampler and $L$ is taken large enough to satisfy $\sum_{k=1}^{L} u_k^{(l)} \prod_{j=1}^{k-1}(1 - u_j^{(l)}) \leq 1 - 10^{-3}$ across all iterations $l$. Then we sweep through the posterior conditionals

$$\left\{\underline{u}_{(1)}, \left(\alpha, \underline{u}_{(2)}\right), \underline{\phi}_{(1)}, \left(\underline{\beta}, \underline{\phi}_{(2)}\right), \sigma^2, \underline{g}\right\}.$$

Parameters that are block updated are enclosed in $(\cdot)$. The posterior conditionals converge to a sample from the full posterior of $(\underline{\beta}, \sigma^2, \alpha, \underline{\phi}, \underline{u}, \underline{g})$. We then choose a partition based on $p(\underline{g}|-)$. Since we have a selection of posterior partitions with their associated posterior probabilities $p(\underline{g}|-)$, we have more choice on the final selected partition. This is a feature missing from the other clustering methods we discussed in Section 2.4, where we only have one partition with no measure of uncertainty.

There are a number of ways we could select the final posterior partition based on $p(\underline{g}|-)$. We propose a variation of the *integrated likelihood* ratio which incorporates Maximum A Posteriori Probability (MAP). The idea here is to choose the lower 10th percentile, $\xi$, of the set of posterior null, or one cluster, partition probabilities based

on 1000 NULL datasets. We control the Type I error of this Bayesian method at 10% so that comparisons can be made with other frequentist methods in Section 4.5. Then using a dataset output a set of $C$ posterior partitions such that $\sum_{c=1}^{C} q_c = 1 - 10^{-3}$, where $q_c$ is the posterior probablity for partition $c$ and $q_1 > q_2 > \ldots > q_C$. If there is a null partition in the set of $C$ partitions with $q_c > \xi$ then we choose this as the final, otherwise we choose a partition $c$ based on MAP. Under this selection criterion we relabel the DPMN model as the Dirichlet Process Normal Mixture model for Clustering (DPNMC).

In the next two sections we address some issues with the sampler when $\alpha$ is small in (4.19) and consider some useful convergence diagnostics which will help later in our simulation study where we determining an adequate number of iterations for the sampler.

### 4.3.3 Accurate simulation scheme for $u_{m^*}$

From the $\underline{u}$ posterior in (4.21) we observe that when $\alpha$ is small drawing $u_{m^*}$ from $Beta(r_{m^*} + 1, \alpha)$ could potentially cause a problem. Instead, we re-write the conditional posterior for $\alpha$ as

$$\alpha|- \sim Gamma\left(a + m^*, b - \sum_{k=1}^{m^*-1} \log(1 - u_k) + V\right),$$

where $V = -\log(1 - U_{m^*})$. Then we see that $p(v|-) = Q(v)r(v)$, where $Q(v) = (1 - e^{-v})^{r_{m^*}}$ and $r(v) = e^{-v\alpha}$. Since $Q(v)$ is a c.d.f and $r(v)$ is a p.d.f, we are able to draw samples from $p(v|-)$ by first drawing $X$ from $r(v)$ then $Y$ from $Q(v)$ using a simple rejection technique, see Tocher (1975). We accept the pair if $Y < X$ and use $X$ as the required sample from $p(v|-)$. Since $Y \to \infty$ as $\alpha \to 0$, we have $Q(V) \to 1$. Therefore samples from $p(\alpha|-)$ are drawn by using the following scheme.

1. On any given pass of the sampler, if $\alpha < \xi$ then go to step 2
2. Generate samples from $p(v|-)$ using $X \sim \text{Exp}(\alpha)$ then $Y \sim U[0, 1]$. We then accept $X$ as a draw from $p(v|-)$ if $Q^{-1}(Y) < X$, where $Q^{-1}(y) = -\log(1 - y^{1/r^{m^*}})$. Otherwise we repeat until the condition is satisfied.

We set $\xi = 1.5$ based on 10,000 random samples from $Beta(1, \alpha)$ such that the number of samples where $\alpha < \xi$ is close to 0.

### 4.3.4 Convergence diagnostics

Convergence here refers to the convergence of the Gibbs Sampler, or other MCMC technique, to its stationary distribution. There are two general questions we can ask with regard to convergence:

1. At what point do we know that we have (essentially) converged to the stationary distribution? (That is, how long should our 'burn-in' period be?)

2. After we have reached the stationary distribution, how many iterations will it take to adequately summarize the posterior distribution?

The answers to both of these questions are rather *ad hoc* because the results are only true asymptotically, and we cannot wait for an infinite number of draws. One intuitive and easily implemented diagnostic tool is a traceplot (or history plot) which plots the parameter value at time $t$ against the iteration number. If the model has converged, the traceplot will hover around the mode of the distribution. A clear sign of non-convergence with a traceplot occurs when we observe some trending in the sample space. However, the problem with traceplots is that it may appear that we have converged, but the chain is trapped (for a finite time) in a local region rather than exploring the full posterior. Another possibility is to look at the *autocorrelation*, which refers to a pattern of serial correlation in the chain, where sequential draws of a parameter, say $\alpha$, from the conditional distribution are correlated. The reason autocorrelation is important is that when it is high the Gibbs sampler will take a very long time to explore the entire posterior distribution. Note that if the level of autocorrelation is high for a parameter of interest, then a traceplot will be a poor diagnostic for convergence. Typically, the level of autocorrelation will decline with increasing number of lags in the chain (e.g. as we go from the 1000th to the 1010th lags). When this dampening does not occur, then we need to re-parameterize the model, as we did in Section 4.3.2, to remove the dependence between the $\alpha$ and $\underline{u}$ using our active and non-active setup.

Other methods to speed up and detect convergence are outlined in Gilks et al. (1995). For our purposes we make use of the Monte Carlo Standard Error (MCSE) and *batching* to diagnose convergence since it is simpler than some of the other proposals to implement and requires less computation. The idea is as follows: suppose we decide to run the chain until the MCSE of the estimated posterior mean of some function $f(\theta)$ of interest is sufficiently small. Here we want the MCSE small in relation to the posterior standard deviation of $f(\theta)$. A rule of thumb is to run the simulation until the MCSE associated with each parameter is less than 5% of the

parameter's posterior standard deviation. So in general for the parametric function $f(\theta)$ for a given run length $N$ and burn-in length $M$ we use batching to estimate $\text{MCSE}(\hat{f}|-)$, where

$$\hat{f} = \frac{\sum_{t=M+1}^{N} f(\theta^t)}{N-M}.$$

To calculate the estimate $\text{MCSE}(\hat{f}|-)$ we use the following steps:

1. Divide the sequence
$$\theta^{M+1}, \ldots, \theta^N$$
into $Q$ equal-length batches of size $L$.

2. Calculate
$$b_q = \frac{1}{L} \sum_{t \in \text{batch}_q} f(\theta^t)$$

3. Check that $b_1, \ldots, b_Q$ are approximately independent. Using the *lag-1* autocorrelation gives an indication of whether batches are approximately independent. If autocorrelation is high, then larger batches are needed.

4. Estimate
$$\text{MCSE}(\hat{f}|-) = \sqrt{\frac{\sum_{q=1}^{Q}(b_q - \bar{b})^2}{Q(Q-1)}},$$
where $\bar{b} = \sum_{q=1}^{Q} b_q/Q$.

## 4.4 Comparison of DPNM with the GP

To understand the reinforcement mechanism acting with a $GG$ as opposed to the standard DPM, Lijoi et al. (2007) considered an extreme setup where the data is far away from the prior. We provide a simulation study similar to that of Lijoi et al. (2007), but add in the DPNM for comparison purposes. In Lijoi et al. (2007) simulation they consider a uniform mixture of three normal distributions with means -4, 0 and 8, and unit variance. They then simulate 100 values from such a mixture and use the data to compare performance against three different mixture models: the DPM model, the mixture of $GG(\beta = 24, \sigma = 0.5)$, and $GG(\beta = 2.23, \sigma = 0.75)$ processes, see Section 3.6. In addition to these three models we also consider the DPNM in our simulation with vague priors on $\mu_0$, $k_0$ and $\sigma^2$ by setting their hyperparameters accordingly. The difference between the DPM in Lijoi et al. (2007) and our DPNM is that they do not place priors on the hyperparameters for $G_0$ and

| $k$ | $DPNM$ ($a = 0.01, b = 0.01/39.13$) | $DPNM$ ($a = 0.001, b = 0.001/39.13$) | $DPM$ ($\alpha = 39.13$) | $GG$ ($\beta = 24, \sigma = 0.5$) | $GG$ ($\beta = 2.23, \sigma = 0.75$) |
|---|---|---|---|---|---|
| 3 | 0.28406 | 0.27400 | 0.00205 | 0.06660 | 0.42490 |
| 4 | 0.24756 | 0.25689 | 0.01295 | 0.19095 | 0.36055 |
| 5 | 0.16522 | 0.18878 | 0.04000 | 0.25175 | 0.15555 |
| 6 | 0.10272 | 0.11044 | 0.08210 | 0.22095 | 0.04575 |
| 7 | 0.06644 | 0.06878 | 0.13690 | 0.14305 | 0.01090 |
| 8 | 0.04228 | 0.04256 | 0.16560 | 0.07395 | 0.00195 |
| 9 | 0.02844 | 0.02294 | 0.16450 | 0.03530 | 0.00035 |
| 10 | 0.01356 | 0.01267 | 0.14395 | 0.01100 | 0.00005 |
| 11 | 0.01644 | 0.00889 | 0.10725 | 0.00455 | - |
| $\geq 12$ | 0.03161 | 0.01406 | 0.14470 | 0.0190 | - |

Table 4.1: Posterior distribution on the number of clusters $k$ arising from the four mixture models centred such that the prior expected number of clusters is 50

$\sigma^2$, but instead estimate them. With the DPNM we set $v_0 = 10^{-3}$ and $\sigma_0^2 = 1$ as the hyperparameters for $\sigma^2$, along with $\mu_1^* = 1$, $v_1 = 10^{-2}$, $\sigma_1^2 = 1$, $\sigma_2^2 = 10^3$ as hyperparameters for $G_0$. In all four setups the expected number of clusters amongst the 100 samples values is set to 50. Thus we see that the prior opinion is far from the truth to highlight the reinforcement mechanism. Under this setup the corresponding parameter values for $\alpha = 39.13$ for the DPM, $GG(\beta = 24, \sigma = 0.5)$ and $GG(\beta = 2.23, \sigma = 0.75)$ for the generalized gamma model. With DPNM since we have a prior on $\alpha$, see (4.22), we fixed $(a, b)$ such that $\mathbb{E}[\alpha] = 39.13$ but our prior belief is fairly vague. Under each setup we simulated results based on 20000 iterations with 2000 burn-in sweeps. Table 4.1 reports the posterior probabilities on the number of clusters. As we see, the performance of $GG(\beta = 2.23, \sigma = 0.75)$ is superior to the other models in terms of recovering the implanted clusters. However, in relation to our DPNM the improvement is only marginal, thus highlighting that not having a prior on $\alpha$ with DPM is rather restrictive and clearly reduces the reinforcement learning ability. Lijoi et al. (2007) extend their $GG$ model by putting a prior on $\sigma$, which causes a marked improvement in performance. However, they focused their prior on up to 100 clusters, which is closer to the truth than having a non-constrained prior as with DPNM.

## 4.5 Comparison of clustering methods

Simulated data depicting Which?'s brand trials, such as the example in Section 1.1.1, enables us to make comparisons between DPNMC and the other clustering meth-

ods described in Section 2.4. The data simulates two of the most common types of product trials at Which?, namely a six or ten brand setup. For each brand we simulate the responses from $t = 20$ different random individuals on a question of interest. The responses are on a 1-5 preference scale (five categories). Since the responses are on a discrete 1-5 scale we take the average $\bar{X}_j$ across all 20 individuals for brand $j$, and under the central limit theorem the $\bar{X}_j$ are approximately normal for large $t$. By using DPNMC to cluster brands at Which? we make a few implicit assumptions. Firstly the within cluster, or response, variance across brands is homogeneous. This is a fair assumption since a fair range of product trials at Which? yield similar response variations by brand. In situations where they differ, an additional respondent factor is included in the model, see Section 7.3. Secondly, we have the same number of raters per brand. Thirdly, the expected number of clusters increases with the number of brands in an approximate logarithmic fashion. Some of these restrictions can be relaxed by extending the DPNM in Section 4.2. For instance we could have a separate response variance per brand. Although the DPMN is an infinite dimensional cluster model some critics would argue its application to clustering at Which? as they ideally seek five classes of product to publish. However, DPMN offers a more formal way to cluster brands using a model based approach and is adaptive, that is it has the ability to learn new classes of products unlike previously seen. The restriction on the ideal five classes is explored further in Chapter 6 by setting appropriate hyperparameters for the $\alpha$ prior through scaling. Later, in Chapter 5, we develop a DP model that closely fits the data using a multinomial distribution. It also offers the ability to control the cluster boundaries that are also commercially viable as opposed to just statistically meaningful through the specification of the $\beta$ parameter in model (5.2), either through simulation or using an integrated likelihood based approach, see Section 5.3.

We simulate three scenarios that are representative of the trials at Which? for six brands, such as the example in Section 1.1.1, and ten brands[1]. The scenarios were ordered such that scenario one had cluster boundaries that were further apart, and closer together as we move towards scenario three. More precisely under scenario one we simulated cluster boundaries where the difference between cluster boundary means was around one. In scenarios two and three we simulated boundary means with differences around 0.8 and 0.5 respectively. We implanted two, three and six clusters in the six brand case, while for the ten we implanted two, five and ten clusters. For example, under six brands we implanted two boundaries, or three clusters,

[1]There are many different types of trials at Which? Most trials consist of less than 20 brands, and more commonly around 6-12

where the first cluster consisted of two brands with responses simulated with higher weights, around 43.5%, in each of the lower two categories (1 or 2). The remaining categories each receiving 4.3%. Similarly in cluster two we generated from the middle category (3) with higher weight, around 71.4%, with the remaining categories taking 7.1% each. Finally in the last cluster, more weight was placed on the top two responses (4-5), around 43.5% in each, with the remaining categories each receiving 4.3%. For convenience, in the six brand case we write $(\underline{X}_1, \underline{X}_2)$ generated with, $W_6C_1 = (43.5\%, 43.5\%, 4.3\%, 4.3\%, 4.3\%)$ for the response weights in the first cluster. Here the notation $W_mC_g$ signifies the category weights for $m$ brands under the $g$th implanted cluster. Similarly $(\underline{X}_3, \underline{X}_4)$ and $(\underline{X}_5, \underline{X}_6)$ were generated with $W_6C_2 = (7.1\%, 7.1\%, 71.4\%, 7.1\%, 7.1\%)$ and $W_6C_3 = (4.3\%, 4.3\%, 4.3\%, 43.5\%, 43.5\%)$ for clusters two and three respectively. More generally, we can re-write, say, $(\underline{X}_1, \underline{X}_2)$ generated with $W_6C_1 = (43.5\%, 43.5\%, 4.3\%, 4.3\%, 4.3\%)$ as $W_6C_1 = (\psi, \psi, 1, 1, 1)$, where the elements are normalised to add to one. Since the methods we compare, apart from DPMMC in Chapter 5, use sample means as inputs some notion of the true mean per cluster is needed. We list the generation weights across all setups along with an estimate of their corresponding true cluster mean as follows:

Six brands - two clusters

1. $\left(\underline{X}_{1,2,3}\right)$ generated with $W_6C_1 = (\psi, \psi, \psi, 1, 1)$
   and cluster mean $(6\psi + 9)/(3\psi + 2)$

2. $\left(\underline{X}_{4,5,6}\right)$ generated with $W_6C_2 = (1, 1, \psi, \psi, \psi))$
   and cluster mean $(12\psi + 3)/(3\psi + 2)$

Six brands - three clusters

1. $\left(\underline{X}_{1,2}\right)$ generated with $W_6C_1 = (\psi, \psi, 1, 1, 1)$
   and cluster mean $(3\psi + 12)/(2\psi + 3)$

2. $\left(\underline{X}_{3,4}\right)$ generated with $W_6C_2 = (1, 1, \psi, 1, 1)$
   and cluster mean $(3\psi + 12)/(\psi + 4)$

3. $\left(\underline{X}_{5,6}\right)$ generated with $W_6C_3 = (1, 1, 1, \psi, \psi)$
   and cluster mean $(9\psi + 6)/(2\psi + 3)$

Six brands - six clusters

1. $(\underline{X}_1)$ generated with $W_6C_1 = (\psi, 1, 1, 1, 1)$
   and cluster mean $(\psi + 14)/(\psi + 4)$

2. $(\underline{X}_2)$ generated with $W_6C_2 = (\psi/2, \psi/2, 1, 1, 1)$
   and cluster mean $(3\psi/2 + 12)/(\psi + 3)$

3. $(\underline{X}_3)$ generated with $W_6C_3 = (1, 1, \psi, 1, 1)$
   and cluster mean $(3\psi + 12)/(\psi + 4)$

4. $(\underline{X}_4)$ generated with $W_6C_4 = (1, 1, \psi/2, \psi/2, 1)$
   and cluster mean $(7\psi/2 + 8)/(\psi + 3)$

5. $(\underline{X}_5)$ generated with $W_6C_5 = (1, 1, 1, \psi/2, \psi/2)$
   and cluster mean $(9\psi/2 + 6)/(\psi + 3)$

6. $(\underline{X}_6)$ generated with $W_6C_6 = (1, 1, 1, 1, \psi)$
   and cluster mean $(5\psi + 10)/(\psi + 4)$

Ten brands - two clusters

1. $\left(\underline{X}_{1,2,3,4,5}\right)$ generated with $W_{10}C_1 = (\psi, \psi, \psi, 1, 1)$
   and cluster mean $(6\psi + 9)/(3\psi + 2)$

2. $\left(\underline{X}_{6,7,8,9,10}\right)$ generated with $W_{10}C_2 = (1, 1, \psi, \psi, \psi))$
   and cluster mean $(12\psi + 3)/(3\psi + 2)$

Ten brands - five clusters

1. $\left(\underline{X}_{1,2}\right)$ generated with $W_{10}C_1 = (\psi, 1, 1, 1, 1)$
   and cluster mean $(\psi + 14)/(\psi + 4)$

2. $\left(\underline{X}_{3,4}\right)$ generated with $W_{10}C_2 = (1, \psi, 1, 1, 1)$
   and cluster mean $(2\psi + 13)/(\psi + 4)$

3. $\left(\underline{X}_{5,6}\right)$ generated with $W_{10}C_3 = (1, 1, \psi, 1, 1)$
   and cluster mean $(3\psi + 12)/(\psi + 4)$

4. $\left(\underline{X}_{7,8}\right)$ generated with $W_{10}C_4 = (1, 1, 1, \psi, 1)$
   and cluster mean $(4\psi + 11)/(\psi + 4)$

5. $\left(\underline{X}_{9,10}\right)$ generated with $W_{10}C_5 = (1, 1, 1, 1, \psi)$
   and cluster mean $(5\psi + 10)/(\psi + 4)$

Ten brands - ten clusters

1. $(\underline{X}_1)$ generated with $W_{10}C_1 = (\psi, 1, 1, 1, 1)$
   and cluster mean $(\psi + 14)/(\psi + 4)$

2. ($\underline{X}_2$) generated with $W_{10}C_2 = (\psi/2, \psi/2, 1, 1, 1)$
   and cluster mean $(3\psi/2 + 12)/(\psi + 3)$

3. ($\underline{X}_3$) generated with $W_{10}C_3 = (\psi/3, \psi/3, \psi/3, 1, 1)$
   and cluster mean $(2\psi + 9/(\psi + 2)$

4. ($\underline{X}_4$) generated with $W_{10}C_4 = (1, \psi/2, \psi/2, 1, 1)$
   and cluster mean $(5\psi/2 + 10)/(\psi + 3)$

5. ($\underline{X}_5$) generated with $W_{10}C_5 = (1, 1, \psi, 1, 1)$
   and cluster mean $(3\psi + 12)/(\psi + 4)$

6. ($\underline{X}_6$) generated with $W_{10}C_6 = (1, \psi/3, \psi/3, \psi/3, 1)$
   and cluster mean $(3\psi + 6)/(\psi + 2)$

7. ($\underline{X}_7$) generated with $W_{10}C_7 = (1, 1, \psi/2, \psi/2, 1)$
   and cluster mean $(7\psi/2 + 8)/(\psi + 3)$

8. ($\underline{X}_8$) generated with $W_{10}C_8 = (1, 1, \psi/3, \psi/3, \psi/3)$
   and cluster mean $(4\psi + 3)/(\psi + 2)$

9. ($\underline{X}_9$) generated with $W_{10}C_9 = (1, 1, 1, \psi/2, \psi/2)$
   and cluster mean $(9\psi/2 + 6)/(\psi + 3)$

10. ($\underline{X}_{10}$) generated with $W_{10}C_{10} = (1, 1, 1, 1, \psi)$
    and cluster mean $(5\psi + 10)/(\psi + 4)$

We took values of $\psi$ in the range $(10, 5, 3)$ for Scenarios 1-3 respectively. For each scenario we constructed 100 random datasets under each setup. Performance on the recovered number of clusters for each method was assessed under each setup. More specifically we assessed performance on three measures:

1. $p_1 = \%$ datasets with all clusters recovered

2. The average number of correctly classified clusters in $(100 - p_1)\%$ clusters not completely recovered. That is when we fail to recover all clusters, we consider the % that were correctly classified amongst the recovered.

3. % Completely recovered clusters amongst cases where we had the same number of implanted clusters. That is, sometimes when we implant three clusters and recover three clusters, their cluster boundaries may not match, or the number of brands in each of the three clusters could be different to what we originally

implanted. The purpose of this measure is to enable fair comparisons with KMeansC, since it restricts the user to specify the number of clusters to output prior to analysis.

With DPNMC, for each dataset, we ran the Gibbs sampler for 500 iterations with a 100 burn-in and drew samples from the posterior distribution. Convergence diagnostics using the block method, see Section 4.3.4, showed an acceptable number of iteration was around 1000. However, running for 500 iterations was acceptable as the difference in results from 500 to 1000 was minimal. Rather than treat $\sigma^2$ as random, see model 4.2, we estimate this by the pooled sample variance $\sum_{j=1}^{m} s_j^2/m$ as we have nonnormal data. We set $\mu_1^* = 1$, $v_1 = 10^{-2}$, $\sigma_1^2 = 1$, $\sigma_2^2 = 10^3$ as the hyperparameters for $G_0$. The posterior partitions were used to obtain the most probable partition in light of the data, see Section 4.3.2.

In principle, as we saw in Section 2.4, each clustering method has its own underlying definition of truth so we may unfairly discriminate against some methods when compare them according to another criterion. Therefore to enable a fair comparison across methods, we calibrated each method to 10% misclassification, or $(100 - p_1)\% = 10\%$, in the complete null (one cluster) situation where all brands are from the same cluster. We calibrated the DPNMC as described earlier using an integrated likelihood method. The other methods were calibrated by tuning their relevent parameters to give 10% misclassification, or ten wrongly classified datasets out of the 100 that were not null. The parameter that was used to tune Method of Normal Scores for Clustering (MNSC) was $\alpha$, $\delta$ for False Discovery Rate for Clustering (FDRC), $\pi$ for Tukey's Method for Clustering (TMC) and $k^*$ for Duncan's Bayesian Decision Theoretic Method for Clustering (DBDTMC). With K-means for Clustering (KMeansC) it was impossible to calibrate the misclassification rate to 10% since it required the number of clusters to be prespecified. The development of the third peformance measure was used to address this issue for KMeansC. With G1C we simply ran as is.

With regard to setting the hyperparameters $(a, b)$ in DPNMC we used a similar setup to Navarro et al. (2006), where we set $a = b = 10^{-2}$ to mimic a noninformative prior on $\alpha$. We will review this choice later in Chapter 6. Figures 4.3-4.5 show the performance measures for all methods under the six brands setup, and Figures 4.6-4.8 for ten brands. In addition, we provide the posterior density for $\alpha$, see Figure 4.2, for the six brands (scenario 1 - three clusters) case along with the posterior mean and standard error for the key parameters shows in Table 4.2. From Figure 4.2 it is clear that the posterior $\alpha$ values can take very large, or small, values therefore giving unpredictable behaviour in the posterior expected number of clusters. We return to

| | Prior | | Posterior | |
|---|---|---|---|---|
| *Parameter* | *Mean* | *SD* | *Mean* | *SD* |
| $\alpha$ | 1.00 | 10.00 | 6.57 | 7.04 |
| $\mu_0$ | 1.00 | $\infty$ | 3.13 | 0.93 |
| $k_0$ | 1.00 | 14.14 | 0.84 | 0.63 |

Table 4.2: Summary of the posterior mean and standard deviation for the key parameters in DPNMC under the six brands (scenario 1 - three clusters) case, with $\hat{\sigma^2} = 0.66$.

this later in Chapter 6. Also referring to Table 4.2 we see that the posterior mean for $\alpha$ is a fair bit away from what we would expect, a value close to two, when we have three implanted clusters in the data. However, this can, in part, be explained by our noninformative prior on $\alpha$.

A number of interesting features can be observed from Figures 4.3-4.8. Firstly, it is clear that DPNMC has improved performance towards more, or less, implanted clusters indicating some instability in the $\alpha$ posterior which is close to being improper here. We return to the issue of setting a prior on $\alpha$ in Chapter 6. The improvement is more apparent under more implanted clusters where it performs better in relation to the other methods under the first performance measure. However, it does not work as well with the five cluster case in ten brands nor with three clusters in six brands. MNSC seems to perform remarkably well across nearly all setups except when we have a larger number of implanted clusters. FDRC generally performed the worst across both the six and ten brand cases, however it does better under more implanted clusters. KMeansC had average performance relative to the other methods based on the third performance measure. Due to KMeansC's restrictions, comparisons were not possible under the maximum number of implanted clusters in both the six and ten brand cases. G1C performs well on the first measure for six brands, but is average under ten brands. Additionally, with Figure 4.3, we see a sharp decline in its second performance measure from scenarios 1-2. TMC performs well on the second performance measure, particularly for six brands. As expected across all cases the performance measures generally decreases from scenarios 1-3. The drop is more noticeable going from the second to the third scenario.

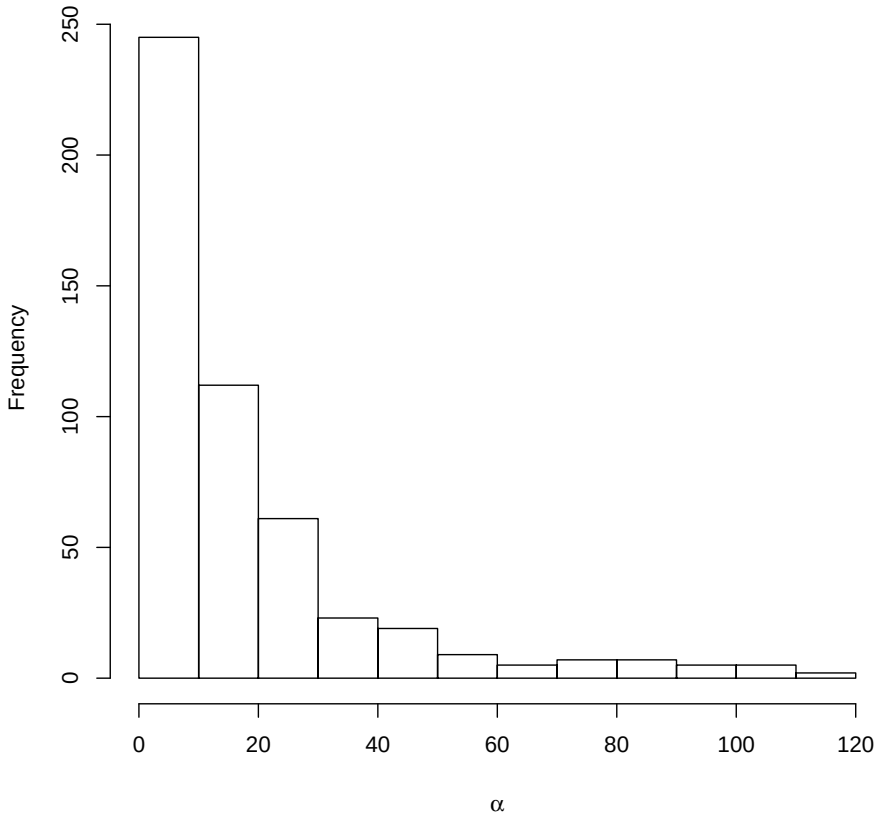


Figure 4.2: Posterior density for $\alpha$ under the six brands (scenario 1 - three clusters) case.

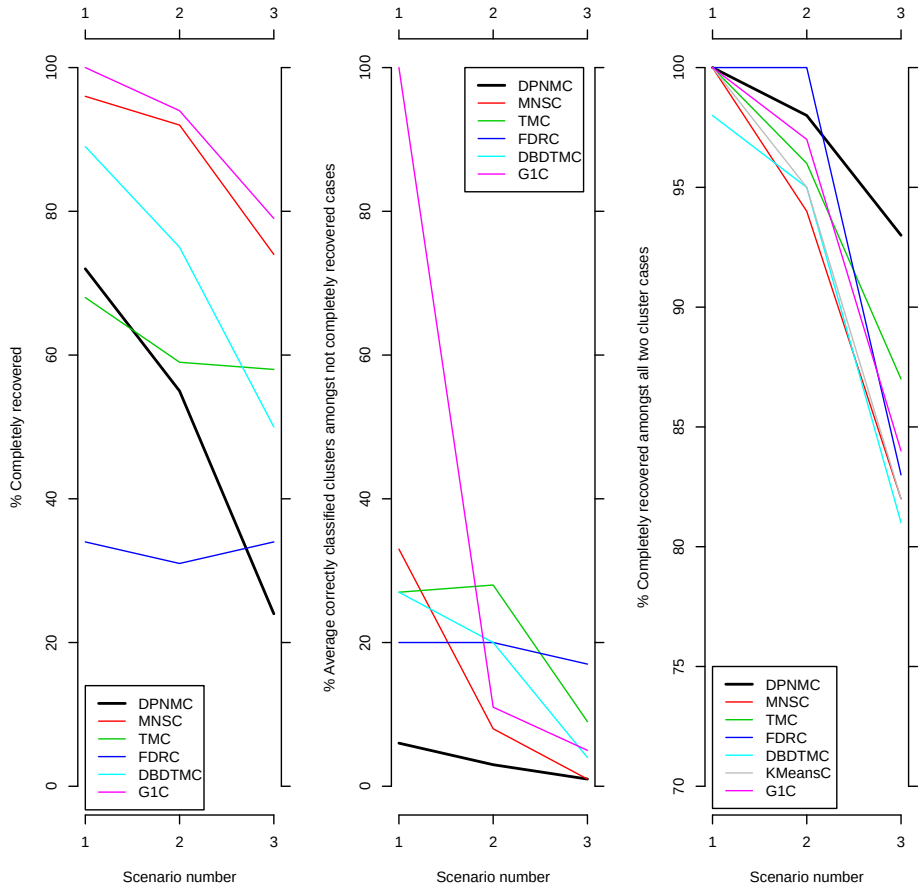


Figure 4.3: Performance of six brands (two implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

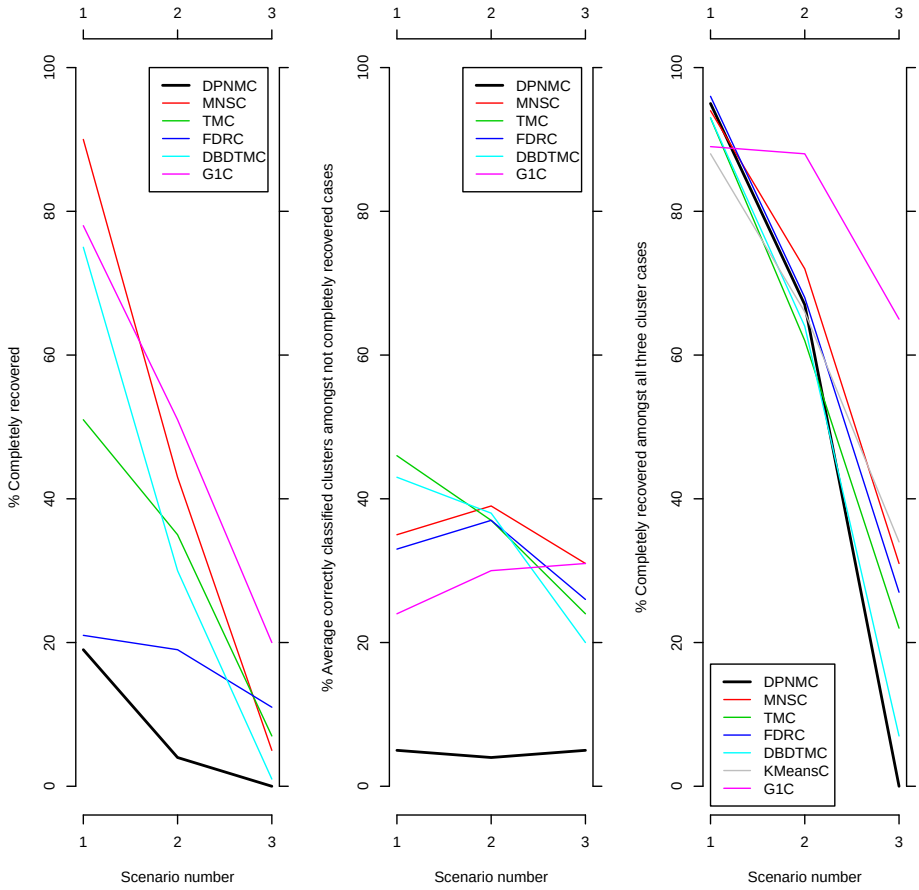


Figure 4.4: Performance of six brands (three implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

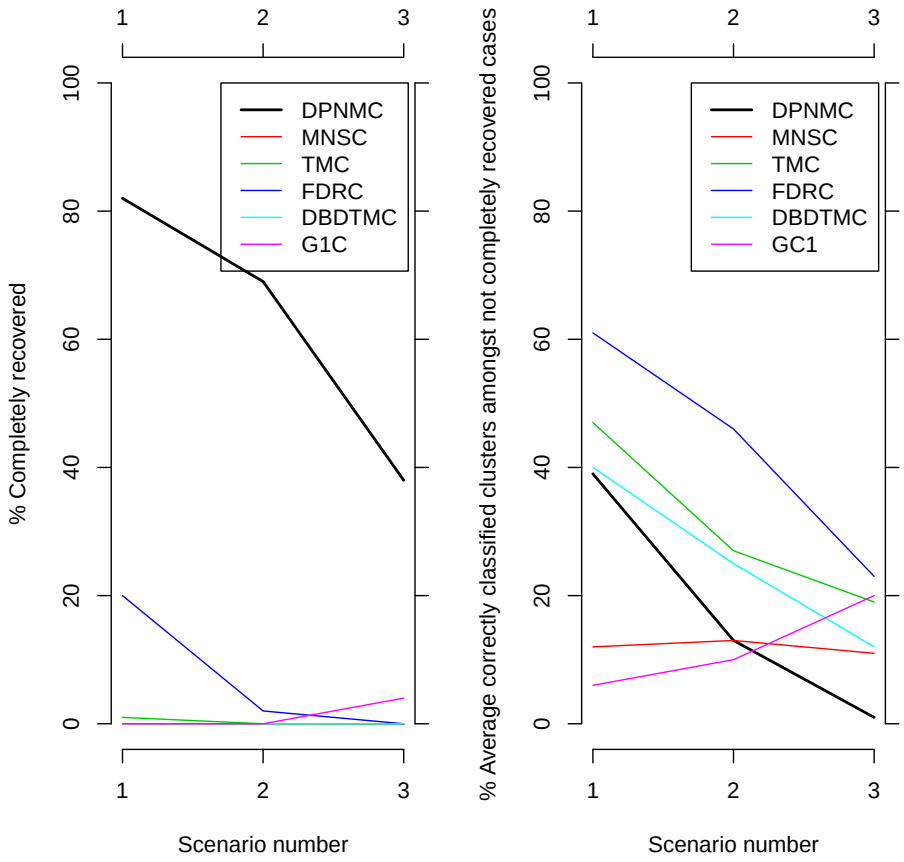


Figure 4.5: Performance of six brands (six implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

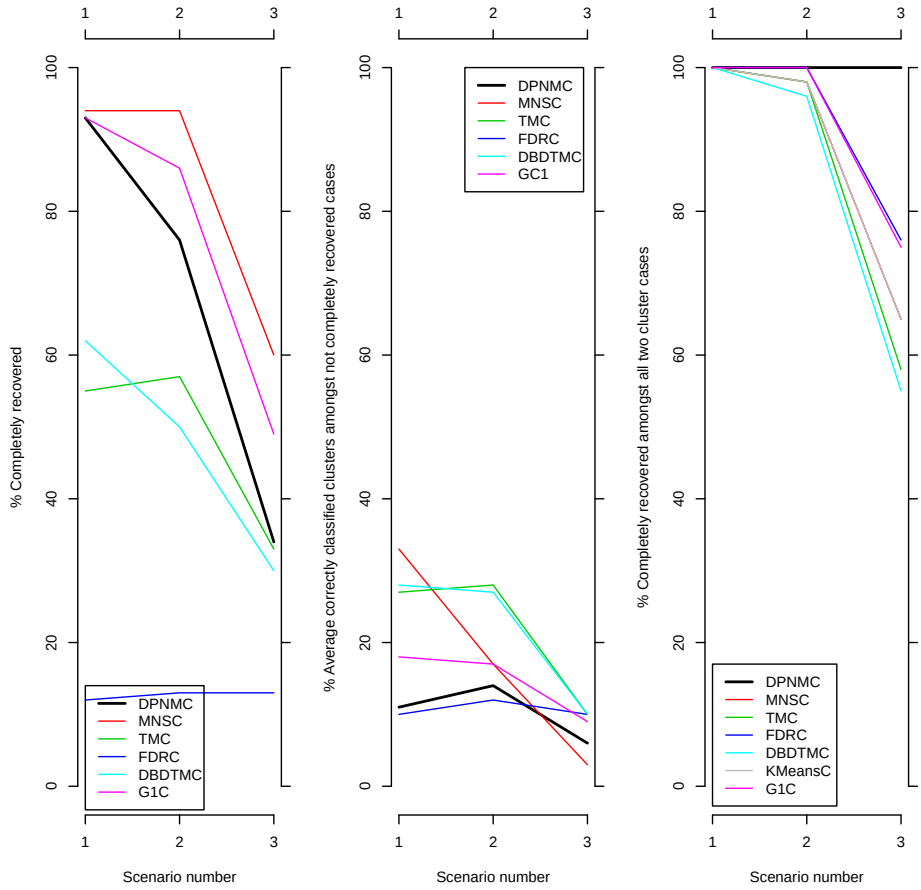


Figure 4.6: Performance of ten brands (two implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

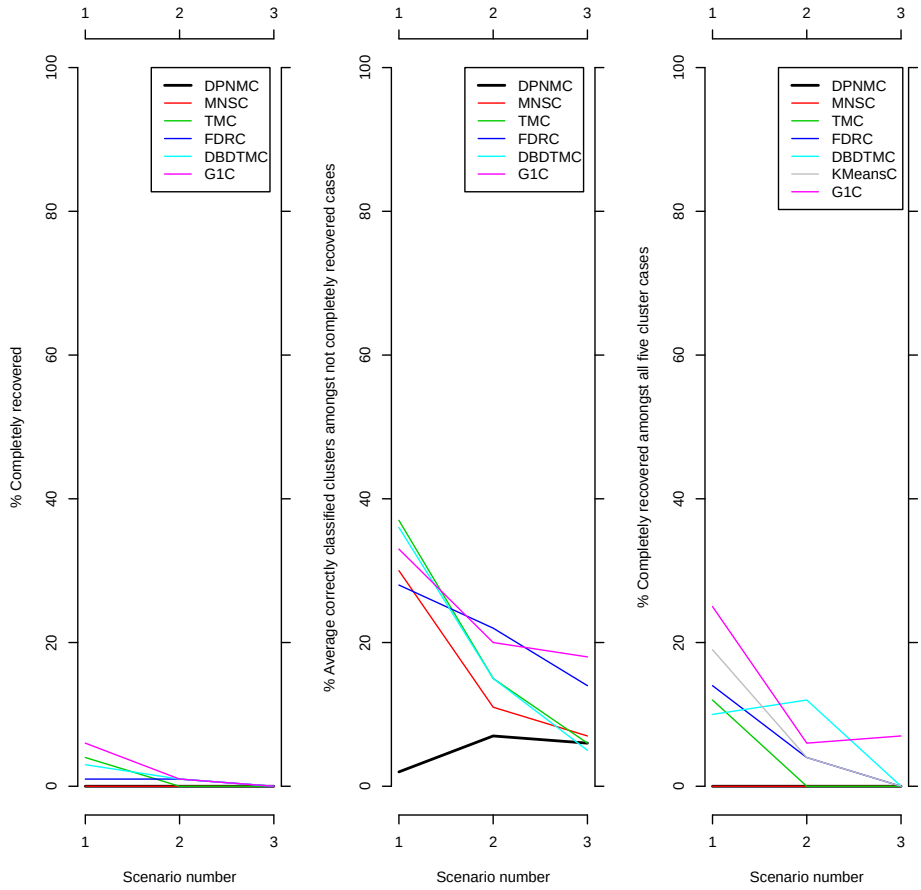


Figure 4.7: Performance of ten brands (five implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

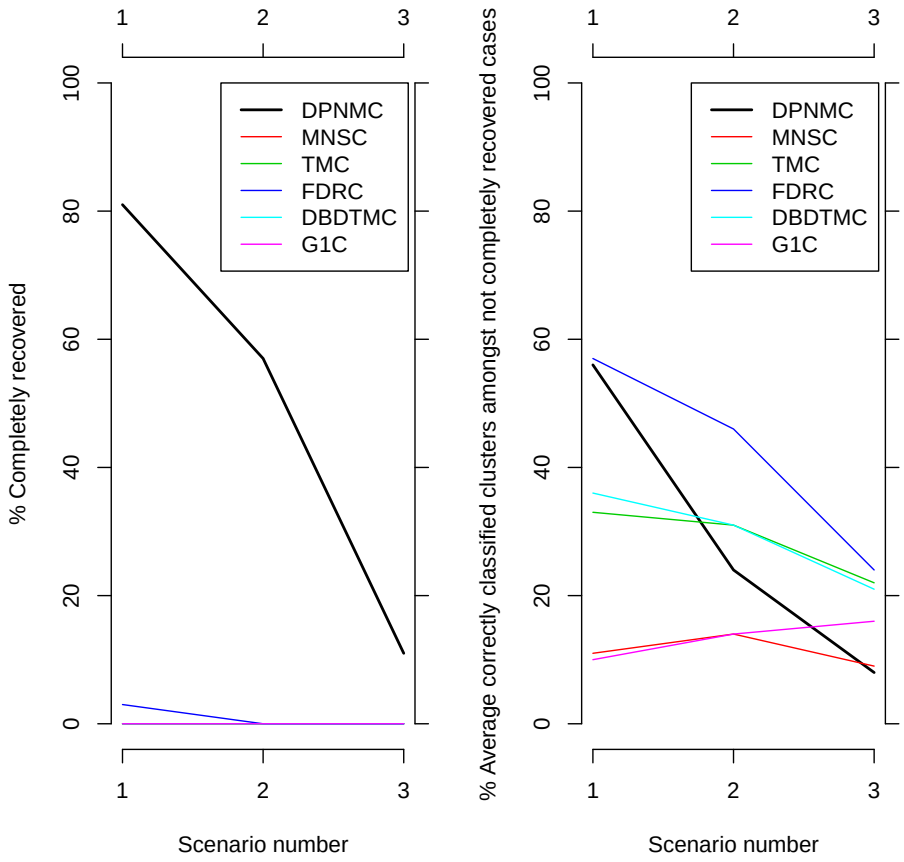


Figure 4.8: Performance of ten brands (ten implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$.

## 4.6 Summary

We have seen that, with Bayesian nonparametrics, two alternative approaches exist to achieve flexibility in clustering:

1. Apply the DPM with a suitable number of hierarchies on the parameters
2. Look at extensions of the DPM model by replacing the DP with a more general prior like the GP.

We demonstrated that the performance gain with the GP was marginal against our DPNM. DPNM was less restrictive than the DPM used by Lijoi et al. (2007) where they showed GPs performance gain over DPM was marked.

We found that the performance of DPNMC in relation to the other clustering method was critically dependent on the specification of the $\alpha$ prior hyperparameters $(a, b)$. Navarro et al. (2006) provides a standard way of setting these by taking them very small. However, as we will see in Chapter 6, this leads to a near-improper posterior and causes unpredictable behaviour in the performance of both DPNMC/DPMMC.

In the next chapter we generalize the DPNMC method to other models and propose a more accurate model for the Which? example discussed in Section 1.1.

# Chapter 5

# Generalization to Non-Normal Data

## 5.1 Introduction

In the previous chapter we demonstrated the implementation of the DPNM for normal data. However, more generally the data can arise from any parametric, or even nonparametric, distribution. Therefore we propose a general framework for the DPM under non-normal data in the next section. We then adapt DPNM to handle multinomial data, which will be particularly useful in proposing an alternative model for clustering brands, see Section 1.1. We conclude with a simulation study, as in Section 4.5.

## 5.2 Generalization

The DPNM (4.2) can be generalized so that the data can occur from any parametric distribution. We now define the unknown parameter vector $\theta = (\gamma, \xi, \beta, \alpha)$, where $\gamma = (\gamma_1, \ldots, \gamma_m)$, and $\gamma_j, \xi$ and $\beta$ can be vectors. The data pdf, or pmf, is $p(X_{ji}|\gamma_j, \xi)$, where $X_{ji}$ is the $i$th replicate for the $j$th object. The $\gamma_j$ are drawn from $G$ where $G$ is drawn from a DP with prior parameters $G_0$ and $\alpha$. We let $G_0$ depend on the parameter $\beta$ and place priors on both $\beta$ and $\alpha$. Finally we place a prior on $\xi$. The

generalized model is summarized below

$$\begin{aligned} X_{ji}|\gamma,\xi &\sim p(X_{ji}|\gamma_j,\xi) \\ \gamma_j|G &\sim G(\cdot) \\ \xi &\sim p(\xi) \\ G|G_0,\alpha &\sim DP(G_0,\alpha) \\ \alpha &\sim p(\alpha) \\ G_0|\beta &\equiv G_0(;\beta) \\ \beta &\sim p(\beta). \end{aligned} \tag{5.1}$$

As in model (4.4) we see that the joint posterior using the stick-breaking construction, see Section 3.5, to sample a realization $G$ from a DP, where $G$ consists of $\underline{\phi}$ and $\underline{u}$ components prior to sampling, can be written as

$$p(\beta,\xi,\alpha,\underline{\phi},\underline{u},\underline{g}|\underline{X}) \propto p(\beta)p(\xi)p(\alpha)p(\underline{u}|\alpha)p(\underline{\phi}|\beta)p(\underline{g}|\underline{u})p(\underline{X}|\underline{g},\underline{\phi},\xi).$$

The full conditionals are as follows

$$p(\underline{u}_{(1)}|-) \propto p(\underline{u}_{(1)}|\alpha)p(\underline{g}|\underline{u}_{(1)}),$$

where $\underline{u} = (u_1,\ldots)$ and $\underline{\phi} = (\phi_1,\ldots)$ have been split into their active and non active parts, see Section 4.3.2. Next consider $p(\alpha,\underline{u}_{(2)}|-)$ by first drawing from

$$p(\alpha|\underline{u}_{(1)}) \propto p(\alpha)p(\underline{u}_{(1)}|\alpha)$$

followed by

$$p(\underline{u}_{(2)}|-) \propto p(\alpha)p(\underline{u}_{(2)}|\alpha).$$

If $p(\underline{\phi}|\beta)$ is a conjugate prior for the likelihood $p(X_{ji}|\gamma,\xi)$, then $p(\underline{\phi}_{(1)k}|-)$ will have the same distributional form as the prior, with updated hyperparameters from both the prior and likelihood.

Similarly, if the prior $p(\beta)$ is conjugate to $p(\phi_{(1)}|\beta)$, then $p(\beta|-)$, will have the same form as the prior, with updated hyperparameters from both $p(\beta)$ and $p(\phi_{(1)}|\beta)$. As there is no contribution from $p(X_{ji}|\gamma,\xi)$ when $g_{jk} = 0$, we see that $p(\phi_{(2)}|-) =$

$p(\phi_{(2)}|\beta)$.

Next if $p(\xi)$ is a conjugate prior for $p(X_{ji}|\gamma,\xi)$, then $p(\xi|-)$ will have the same form as the prior, with updated hyperparameters from both $p(\xi)$ and $p(X_{ji}|\gamma,\xi)$.

Finally

$$\begin{aligned} p(\underline{g}|-) &\propto \prod_{j=1}^{m}\prod_{i=1}^{t}\prod_{k=1}^{L} p(g_{jk}|\underline{u})\left\{p(X_{ji}|\phi_k,\xi)\right\}^{g_{jk}} \\ &= \prod_{k=1}^{L} w_k^{r_k}\left\{\prod_{j=1}^{m}\prod_{i=1}^{t} p(X_{ji}|\phi_k,\xi)\right\}^{g_{jk}} \\ &= \prod_{j=1}^{m}\prod_{k=1}^{L}\tilde{w}_{jk}^{g_{jk}}, \end{aligned}$$

where

$$\tilde{w}_{jk} = \frac{w_k\prod_{i=1}^{t} p(X_{ji}|\phi_k,\xi)}{\sum_{k'=1}^{L} w_{k'}\prod_{i=1}^{t} p(X_{ji}|\phi_{k'},\xi)}.$$

The sampler is now implemented as described in Section 4.3.2 (replacing $\sigma^2$ by $\xi$).

## 5.3 Modelling discrete data with an infinite number of clusters

Thus far with DPNM, see Section 4.3.2, we have considered the responses, $X_{ji}$, to occur on a continuous scale. However, our example in Section 1.1, the response for an object attribute question is on an $s$ point ordered preference scale, where 1 is low and $s$ is high preference. Here we can define object $j$'s binary response by individual $i$ on an $s$ point scale by $\underline{X}_{ji} = (X_{ji1},\ldots,X_{jis})$. By using the DPM we are assuming that each object belongs to one of an infinite number of clusters. Then $\underline{X}_{ji}$ is multinomial with parameters $\underline{\theta}_j = (\theta_{j1},\ldots,\theta_{js})$, where $\theta_{jl}$ denotes the probability with which the $j$th object had rating $l$. More conveniently, we can represent the $\theta_{jl}$ in terms of cluster indicator variables $g_{jk}$, by writing $\theta_{jl} = \prod_{k=1}^{L}\phi_{kl}^{g_{jk}}$. Since the DD, see (3.5), is conjugate to the multinomial, we assign the base distribution $G_0$ as a DD. However, the DD is rather restrictive here. In reality, using our example in Section 1.1, we see that it is unrealistic to assume the responses across all $s$ categories were skewed in one direction. In some product tests brands often concentrate either at the top, or bottom end 'Budget buys' of the market, we are more likely to observe responses that are either concentrated towards the upper, or lower, end of

the preference scale. In contrast with mixed brand trials we can assume that the response will fall into one of the five categories with equal probability. Some of the response variations across various DDs are shown in Figure 3.1. We accommodate this by using a mixture of DDs (MDD) for $G_0$, where the mixtures will represent the $R$ most likely profiles, with associated probabilities $\rho_r, r = 1, \ldots, R$. The set of profile weights for the $r$th profile is denoted by $\underline{a}^r = (a_1^r, \ldots, a_s^r)$, where $\sum_{l=1}^{s} a_l^r = 1$. We introduce a profile indicator

$$z_{kr} = \begin{cases} 1 & ; \text{if the } k\text{th cluster takes on the } r\text{th profile} \\ 0 & ; \text{o.w,} \end{cases}$$

for $r = 1, \ldots, R$, so that $P(z_{kr} = 1|\underline{\rho}) = \rho_r$. Under this revision model (5.1) becomes

$$\begin{aligned} \underline{X}_{ji}|\underline{\theta}_j &\sim Mult(1, \underline{\theta}_j) \\ \underline{\theta}_j|G &\sim G(\cdot) \\ G|G_0, \alpha &\sim DP(G_0, \alpha) \\ \alpha|a, b &\sim Gamma(a, b) \\ G_0|\beta, \underline{a} &= DD(\beta \underline{a}) \\ \underline{a}|\underline{\rho} &\sim \sum_{r=1}^{R} \rho_r \delta(\cdot, \underline{a}^r) \\ \underline{\rho} &\sim DD(e^* \underline{q}), \end{aligned} \tag{5.2}$$

where $\underline{a}$ is a matrix with rows $\underline{a}^r$. For simplicity we shall take the parameter $\beta > 0$ to be fixed. Here $\beta$ is a precision parameter for the $\underline{\phi}_k$. We observe some differences between the revised model (5.2) and the previous (4.2). Firstly, $\underline{\phi}_{(1)}$ is now an $m^*$ by $s$ matrix, where cluster $k$th row vector has distribution $G_0$, and $\underline{\phi}_{(2)}$ an $L - m^*$ by $s$ matrix. Also, $r_k$ now denotes the number of $\underline{\theta}_j$ that are in the $k$th cluster, where $P[\underline{\theta}_j = \underline{\phi}_k] = w_k$. We also introduce $\underline{z}_{(1)}$, which is an $m^*$ by $R$ matrix of active $z_{kr}$, and $\underline{z}_{(2)}$ the $L - m^*$ by $R$ matrix of non-active profiles. Also, since the responses are now taken to be multinomial, there is no additional common level one parameter such as $\sigma^2$. Finally, $\underline{q}$ denotes the prior profile weights, while $e^*$ is the precision parameter for the DD of $\underline{q}$.

We will refer to model (5.2) as the Dirichlet Process Multinomial Mixture (DPMM) model. Based on these revisions, we sample in order, from the following conditional

posterior distributions

$$\begin{aligned}
p(\underline{u}_{(1)}|-) &\propto p(\underline{u}_{(1)}|\alpha)p(\underline{g}|\underline{u}_{(1)}) \\
p(\underline{z}_{(1)k}|-) &\propto p(\underline{z}_{(1)k}|\underline{\rho})p(\underline{\phi}_{(1)k}|\beta,\underline{z}_{(1)k}) \\
p(\underline{\phi}_{(1)k}|-) &\propto p(\underline{\phi}_{(1)k}|\beta,\underline{z}_{(1)k})p(\underline{X}|\underline{g}_k,\underline{\phi}_{(1)k}) \\
p(\alpha|\underline{u}_{(1)}) &\propto p(\alpha)p(\underline{u}_{(1)}|\alpha) \\
u_{(2)k}|- &\sim Beta(1,\alpha) \\
p(\underline{\rho}|\underline{z}_{(1)}) &\propto p(\underline{\rho})p(\underline{z}_{(1)}|\underline{\rho}) \\
p(z_{(2)kr}=1|\underline{\rho}) &= \rho_r \\
\underline{\phi}_{(2)k}|- &\sim DD\left(\beta\sum_{r=1}^{R} z_{kr}\underline{a}^r\right) \\
p(\underline{g}|-) &\propto p(\underline{g}|\underline{u})p(\underline{X}|\underline{g},\underline{\phi}),
\end{aligned}$$

where $\underline{z}_{(1)k}$ denotes the active profile indicator vector for the $k$th cluster, $\underline{\phi}_{(1)k}$ denotes the vector for the $k$th cluster of the active $\underline{\phi}_{(1)}$ and $\underline{g}_k = (g_{1k},\ldots,g_{mk})$. Similarly $\underline{z}_{(2)k}$ and $\underline{\phi}_{(2)k}$ denote the $k$th cluster of the non-active cases. Model (5.2) is illustrated graphically in Figure 5.3.

We now compute the various components above for use in the sampler. We start, as before in Section 4.3.2, by finding $m^*$, then update $u_{(1)k}$ as in equation (4.21). Next, the full conditional of $\underline{z}_{(1)k}$ is

$$p(z_{(1)kr}=1|-) \propto \rho_r \frac{\prod_{l=1}^{s}\phi_{kl}^{\beta a_l^r-1}}{\prod_{l'=1}^{s}\Gamma(\beta a_{l'}^r)},$$

where $\propto$ means proportional to as a function of $r$, depends only on for active $k$, which gives

$$p(\underline{\phi}_{(1)k}|-) \propto \prod_{l=1}^{s}\phi_{kl}^{\beta\sum_{r=1}^{R} z_{kr}a_l^r+\sum_{j=1}^{m} g_{jk}X_{j.l}-1},$$

so that

$$\underline{\phi}_{(1)k}|- \sim DD\left(\beta\sum_{r=1}^{R} z_{kr}a_l^r + \sum_{j=1}^{m} g_{jk}X_{j.l}\right), \tag{5.3}$$

where $X_{j.l} = \sum_{i=1}^{t} X_{jil}$ denote the number of times the $j$th object had the $l$th rating across all responses. It follows that $\underline{X}_{j.}|\underline{\theta}_j \sim Mult(t,\underline{\theta}_j)$. We see that the DD in

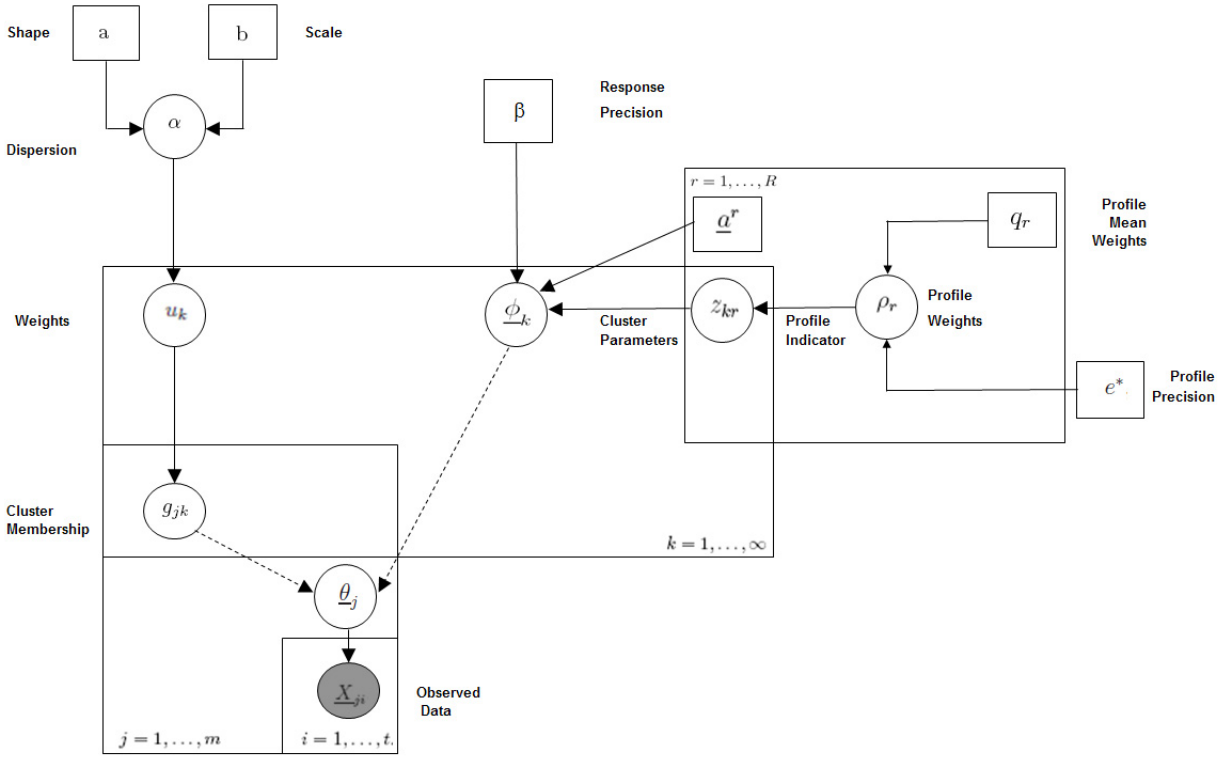


Figure 5.1: Dependencies in the infinite cluster model for discrete data. Shaded circles denote observed variables, white circles are latent variables, squares represent specified hyperparameters, and plates indicate sets of independent replications of the processes shown inside them. Dashed lines indicate the child node is derived from its parent nodes.

(5.3) is based on a weighted mixture of $R$ different profiles along with the data for a particular cluster $k$. The full conditional for $\alpha$ is given by equation (4.22) and

$$u_{(2)k}|- \sim Beta(1, \alpha)$$

as before. Next

$$p(\underline{\rho}|\underline{z}_{(1)}) \propto \prod_{r=1}^{R} \rho_r^{e^* q_r + z_{\cdot r} - 1},$$

so that

$$\underline{\rho}|\underline{z}_{(1)} \sim DD(e^*\underline{q} + \underline{z}_{\cdot}),$$

where $\underline{z}_{\cdot} = (z_{\cdot 1}, \ldots, z_{\cdot R})$, and $z_{\cdot r}$ denotes the number of active clusters that had the $r$th profile. Next $p(z_{(2)kr} = 1|-) = \rho_r$ for non-active $k$ and

$$\underline{\phi}_{(2)k}|- \sim DD\left(\beta \sum_{r=1}^{R} z_{kr}\underline{a}^r\right).$$

Finally

$$p(\underline{g}|-) \propto \prod_{j=1}^{m}\prod_{k=1}^{L} \tilde{w}_{jk}^{g_{jk}},$$

where

$$\tilde{w}_{jk} = \frac{w_k \prod_{l=1}^{s} \phi_{kl}^{X_{j.l}}}{\sum_{k'=1}^{L} w_{k'} \prod_{l'=1}^{s} \phi_{k'l'}^{X_{j.l'}}}.$$

Notice that the non-active full conditional posteriors for $u_{(2)k}$, $\underline{z}_{(2)k}$ and $\underline{\phi}_{(2)k}$ do not involve the data. In a similar way to DPNM, we sweep through the above conditional posteriors in the sequence $\underline{u}_{(1)}, \underline{z}_{(1)}, \underline{\phi}_{(1)}, \left(\alpha, \underline{u}_{(2)}\right), \left(\underline{\rho}, \underline{z}_{(2)}, \underline{\phi}_{(2)}\right), \underline{g}$. Here we update the components in $(\cdot)$ as a block update, which makes the sampler more efficient, as it avoids the ergodicity constraint described in Section 4.3.1. At the end of each sweep of the sample we update the current state of $r_k$ prior to the starting the next sweep. Over time these samples converge to samples from the full posterior distribution of $\underline{u}_{(1)}, \underline{z}_{(1)}, \underline{\phi}_{(1)}, \left(\alpha, \underline{u}_{(2)}\right), \left(\underline{\rho}, \underline{z}_{(2)}, \underline{\phi}_{(2)}\right), \underline{g}$. As with DPNM we see from (4.21) that the last term $u_m^* \sim Beta(r_{m^*} + 1, \alpha)$ causes problems when $\alpha \to 0$. As before we address this using the transformation proposed in Section 4.3.3. Given the complexity in working out the conditional posterior for $\beta$, we estimate it from its marginal likelihood ignoring the DP structure, i.e. as if $\alpha = \infty$. We also ignore the

profile structure of DPMM in this estimate given its complexity, therefore assuming equal weights for all $a_l$. This is sensible when we have no information about the profiles. Since this is a crude estimate for $\beta$, in the next section, we do check its sensitivity in our comparisons. Under these assumptions we see that the integrated likelihood for $\beta$ is

$$\begin{aligned} p(\underline{X}|\beta) &\propto \int p(\underline{X}|\underline{\phi})p(\underline{\phi}|\beta)d\underline{\phi} \\ &\propto \int \prod_{j=1}^{m} \left\{ \prod_{l=1}^{s} \phi_{jl}^{a_l\beta+X_{j.l}-1} \frac{\Gamma(\beta)}{\prod_{l=1}^{s}\Gamma(a_l\beta)} \right\} d\underline{\phi} \\ &= \prod_{j=1}^{m} \left\{ \frac{\prod_{l=1}^{s}\Gamma(a_l\beta+X_{j.l})}{\Gamma(\beta+\sum_{l=1}^{s}X_{j.l})} \frac{\Gamma(\beta)}{\prod_{l=1}^{s}\Gamma(a_l\beta)} \right\}. \end{aligned}$$

Then, taking logs, the integrated log-likelihood of $\beta$ is

$$\begin{aligned} l(\beta) &= \sum_{j=1}^{m} \left[ \sum_{l=1}^{s} \log\{\Gamma(a_l\beta+X_{j.l})\} - \log\left\{ \Gamma\left(\beta+\sum_{l=1}^{s}X_{j.l}\right)\right\}\right] \\ &\quad +m\left[\log\{\Gamma(\beta)\} - \sum_{l=1}^{s}\log\{\Gamma(a_l\beta)\}\right]. \end{aligned}$$

Maximising this function with respect to $\beta$ gives the estimate $\hat{\beta}$, with approximate S.E $\left\{-l''(\hat{\beta})\right\}^{-1/2}$, which is obtained from the Hessian matrix.

## 5.4 Comparison of clustering methods

We revisit the simulation study in Section 4.5, but also add in the performance of the DPMM model. As before with DPNM, we use a variation of the integrated likelihood ratio, see Section 4.3.2, to pick the final partition in DPMM. We label this the Dirichlet Process Multinomial Mixture model for Clustering (DPMMC).

Under each setup for DPMMC we assume five profiles that are realistic of the main types of trials at Which? These are

1. Low budget trials where more focus is placed on cost rather than performance, so more weighting is placed on the lower two responses (1-2). The weights we took were $\underline{a}^1 = (30\%, 30\%, 13.3\%, 13.3\%, 13.3\%)$

2. Mixed brand trials, where there is a variation of brands from the top, middle

and bottom end of the market. Here the weights were $\underline{a}^2 = (20\%, 20\%, 20\%, 20\%, 20\%)$

3. We took weights $\underline{a}^3 = (13.3\%, 13.3\%, 13.3\%, 30\%, 30\%)$ to respresent product trials from the top end of the market, hence the higher weighting on the top two responses (4-5)

4. Trials consisting of brands from the middle market were represented by profile $\underline{a}^4 = (10\%, 10\%, 60\%, 10\%, 10\%)$

5. Finally, the last profile, $\underline{a}^5 = (30\%, 13.3\%, 13.3\%, 13.3\%, 30\%)$, consisted of trials with more brands from the top and bottom end of the market.

Note that we take some account for the ordinal nature of the data using these profiles to model the types of response variation. As we have no prior knowledge about the occurrence of these profiles at Which? we let $\underline{q} = (20\%, 20\%, 20\%, 20\%, 20\%)$ and $e^* = 1$ so that out prior belief on $\mathbb{E}[\rho_r] = 20\%$ with $\mathbb{V}[\rho_r] = 8\%$, therefore allowing some degree of uncertainty around our prior profile weights. With DPMMC, for each dataset, as with DPNMC, we ran the Gibbs sampler for 500 iterations with a 100 burn-in and drew samples from the posterior distribution. To make comparisons fair across methods, we calibrated each method to 10% misclassification, or $(100 - p_1)\% = 10\%$, in the complete null (one cluster) situation where all brands were from the same cluster. We experimented with values of $\beta$ in the range of the approximate 95% interval derived using the integrated likelihood, see previous section, across all cases. We found $\hat{\beta} = 7$ was adequate both in terms of performance as well as depicting the variation between responses similar to that of a standard Which? trial. With regard to setting the hyperparameters $(a, b)$, as before with DPNMC, we use a similar setup to Navarro et al. (2006) where we set $a = b = 10^{-2}$. We will review this choice later in Chapter 6. Figures 5.3-5.5 shows the performance measures for all methods under the six brands setup, and Figures 5.6-5.8 for ten brands. In addition, we provide the posterior density for $\alpha$, see Figure 5.2, for the six brands (scenario 1 - three clusters) case along with the posterior means and standard deviations for key parameters shown in Table 5.1. From Figure 5.2 it is clear, as with DPNMC, that the posterior $\alpha$ values can take very large, or small, values therefore giving unpredictable behaviour in the posterior expected number of clusters. We return to this later in Chapter 6. Also, turning to Table 5.1, it is clear that the posterior SD for $\alpha$ is high as with DPNMC, see Section 4.5, therefore casting more uncertainty around the true posterior expected number of clusters. Notice the higher posterior weight placed on profies 1, 3 and

| | Prior | | Posterior | |
|---|---|---|---|---|
| *Parameter* | *Mean* | *SD* | *Mean* | *SD* |
| $\alpha$ | 1.00 | 10.00 | 3.30 | 4.66 |
| $\rho_1$ | 0.20 | 0.28 | 0.27 | 0.21 |
| $\rho_2$ | 0.20 | 0.28 | 0.10 | 0.17 |
| $\rho_3$ | 0.20 | 0.28 | 0.23 | 0.21 |
| $\rho_4$ | 0.20 | 0.28 | 0.33 | 0.22 |
| $\rho_5$ | 0.20 | 0.28 | 0.08 | 0.14 |

Table 5.1: Summary of the posterior mean and standard deviation for the key parameters in DPMMC under the six brands(three clusters) case.

4, which coincides with our simulated clusters, i.e. two from bottom, two from top and two from the middle market respectively, although the posterior SDs are still quite large.

A number of interesting features can be observed from Figures 5.3-5.8. Firstly, it is clear that DPMMC performs better in relation to the other methods under more implanted clusters, particularly with ten brands. In relation to DPNMC, DPMMC performs better under fewer implanted clusters and about the same with more. FDRC performed the worst across both the six and ten brand cases. When comparisons were made with KMeansC under the third performance measure, DPMMC performs better across most cases. However, under such cases, it is average on performance measure one. As before, due to KMeansCs restrictions, comparisons were not possible under the maximum number of implanted clusters for both the six and ten brand cases. As before, G1C performs well on the first measure for six brands, but is average under ten brands. DPMMC shows improvement on the second performance measure, particularly in relation to DPNMC, as seen from the figures. As before, across all cases the performance measures generally decrease from scenarios 1-3. Again, the drop is more noticeable going from the second to the third scenario.

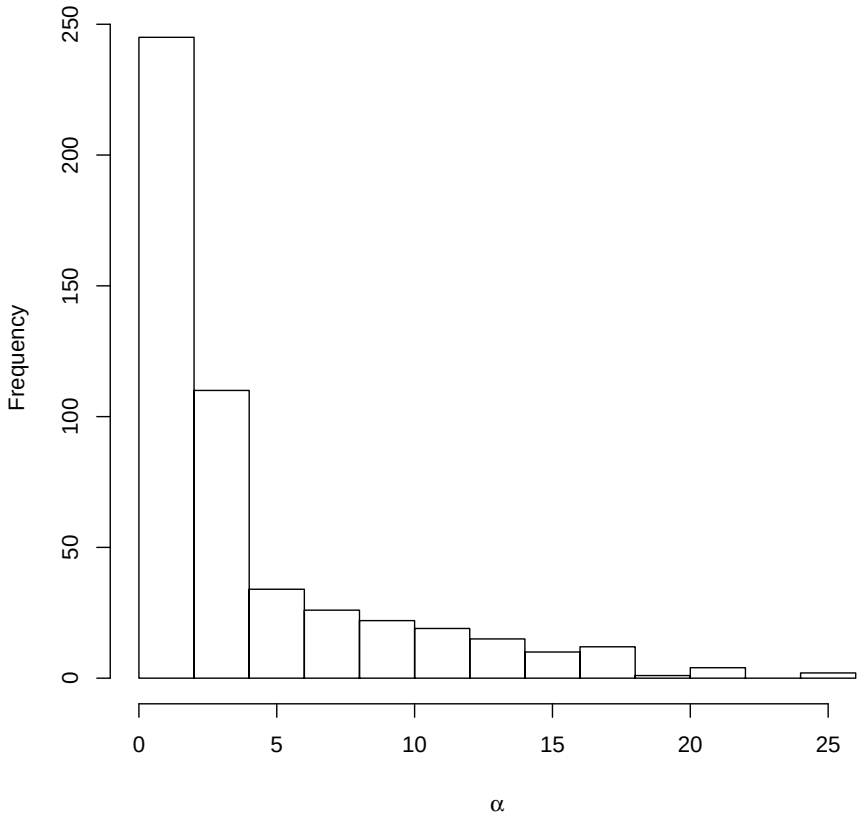


Figure 5.2: Posterior density for $\alpha$ under the six brands (scenario 1 - three clusters) case.

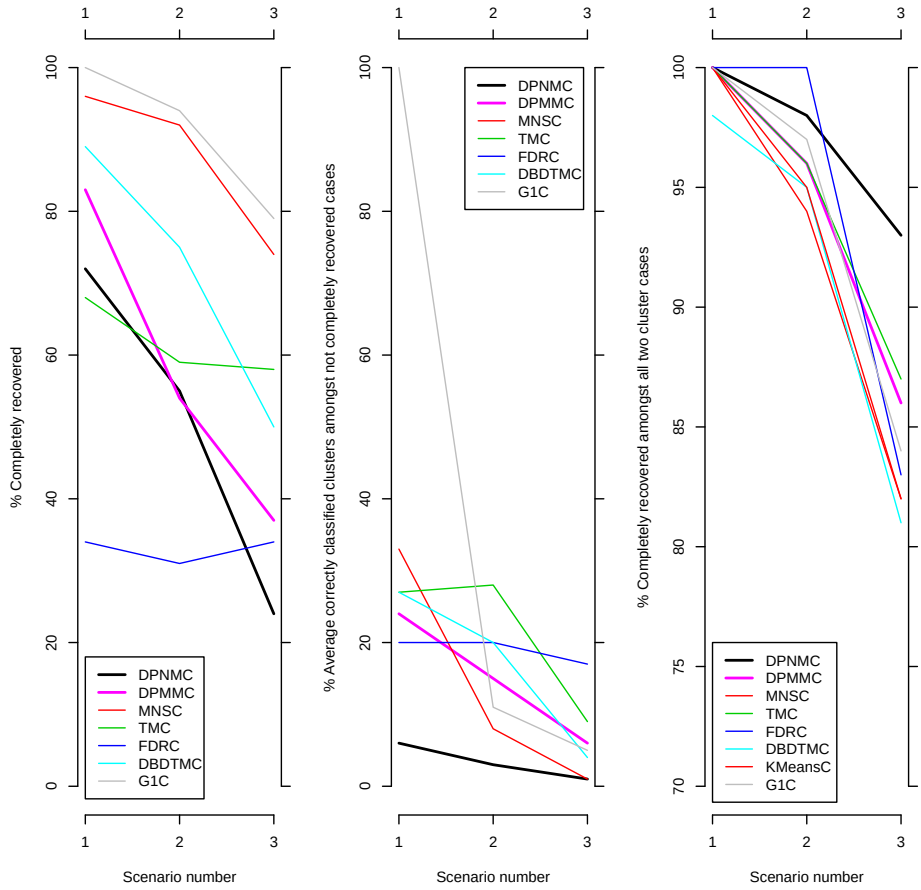


Figure 5.3: Performance of six brands (two implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

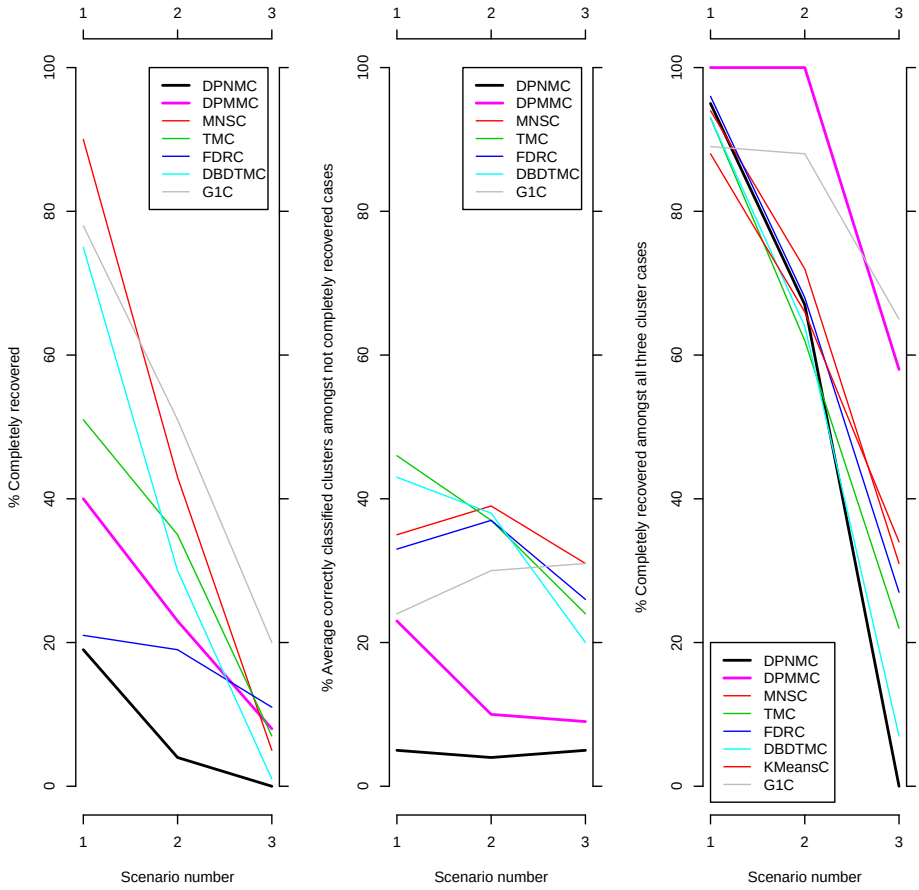


Figure 5.4: Performance of six brands (three implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

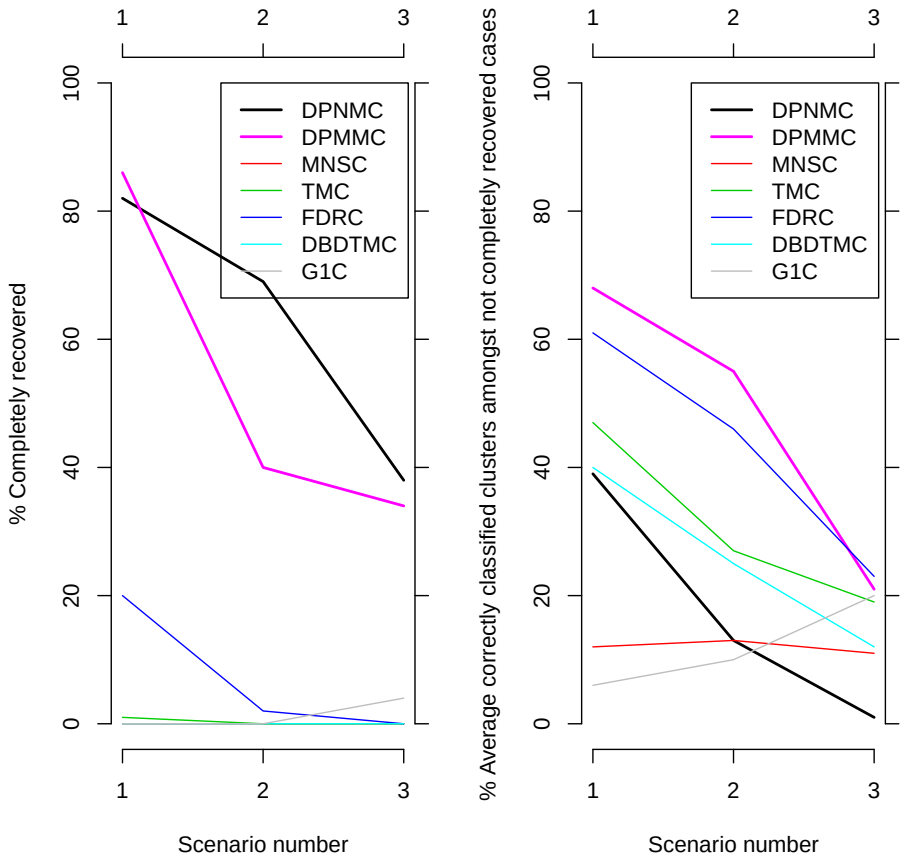


Figure 5.5: Performance of six brands (six implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

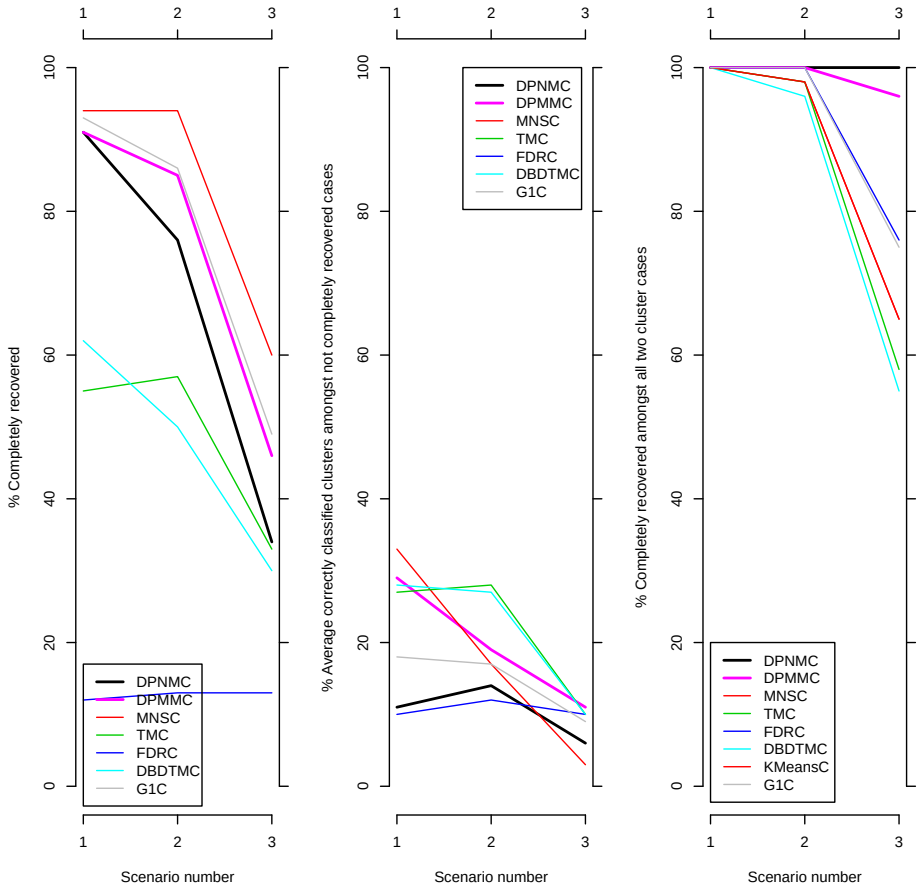


Figure 5.6: Performance of ten brands (two implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

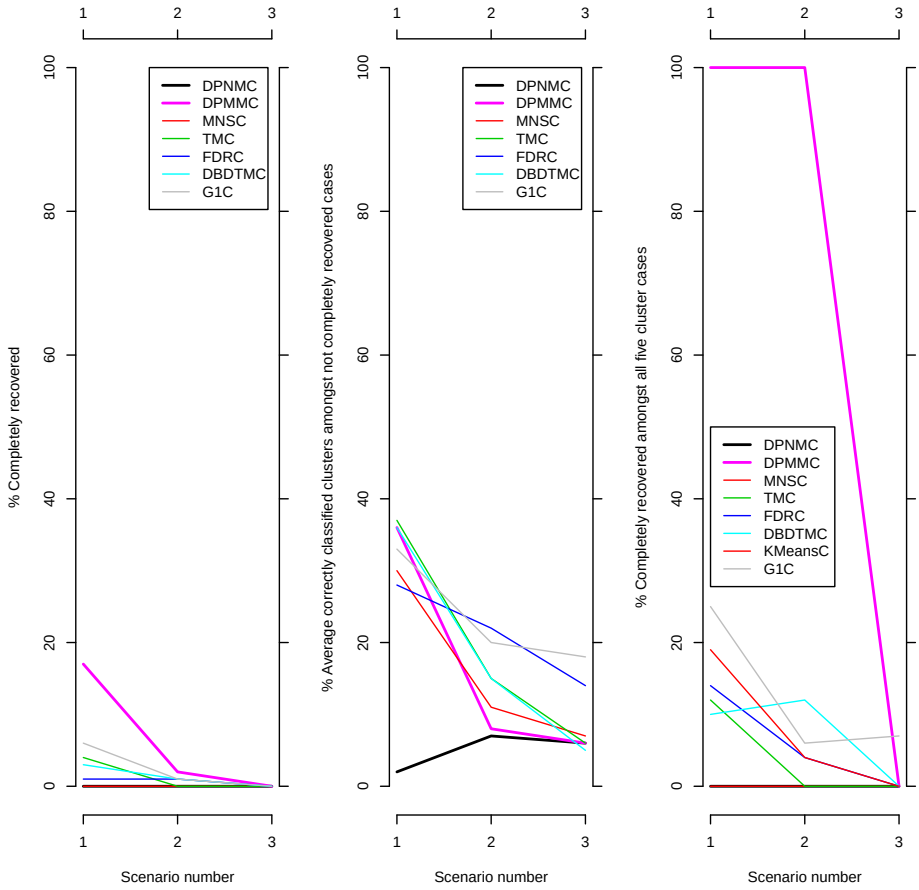


Figure 5.7: Performance of ten brands (five implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

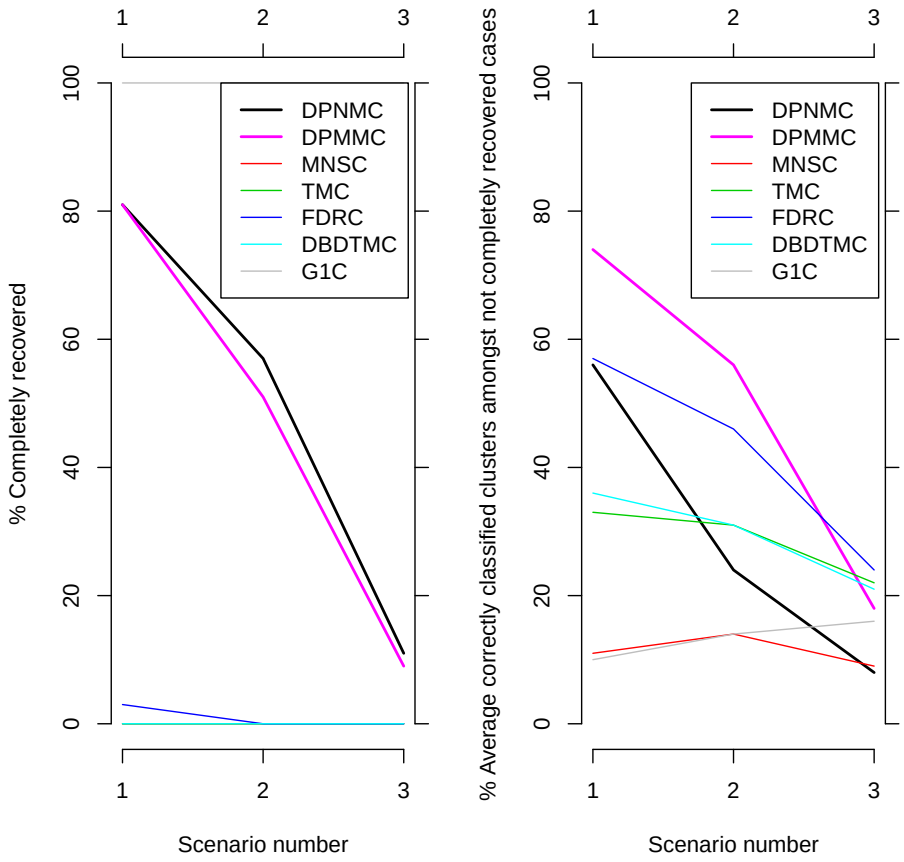


Figure 5.8: Performance of ten brands (ten implanted clusters). The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 10^{-2}$, $b = 10^{-2}$ and $\hat{\beta} = 7$.

## 5.5 Comparison of marginal and conditional methods

Thus far we have demonstrated the implementation of a DPM using the conditional method, see Section 4.3.1. We now consider an implementation using the marginal method given in Navarro et al. (2006) to handle discrete data, which is essentially DPMM but with one profile under the marginal method. In Navarro et al. (2006) they consider the posterior of $(\alpha, \underline{g}, \underline{\phi})$. Starting with the posterior for $\alpha$, as before with DPMM, we let the prior for $\alpha \sim Gamma(a, b)$. Antoniak (1974) observed that the posterior distribution for $\alpha$ is influenced only by the number of distinct clusters $n$, and not by the details of the allocation of observations to those clusters. The probability that $n$ clusters will be observed in $m$ samples is

$$p(n|\alpha, m) = \frac{\Gamma(\alpha)}{\Gamma(\alpha + m)} z_{mn} \alpha^n, \tag{5.4}$$

where $z_{mn}$ is an unsigned Stirling number of the first kind, see Antoniak (1974). To compute $z_{mn}$ we make use of the following recursive relations for the signed Stirling numbers of the first kind

$$\begin{aligned} z^*_{m1} &= (-1)^{m-1}\Gamma(m) \\ z^*_{mn} &= z^*_{(m-1)(n-1)} - (n-1) z^*_{(m-1)(n)}, 1 < n \leq m, \end{aligned}$$

where $1 < n \leq m$. We see that the posterior distribution for $\alpha$ given $n$ and $m$ is

$$\begin{aligned} p(\alpha|n, m) &\propto p(n|\alpha, m) p(\alpha) \\ &\propto B(\alpha, m) \alpha^{a+n-1} e^{-b\alpha}, \end{aligned} \tag{5.5}$$

where $B(\cdot, \cdot)$ is the standard beta function. By expanding $B(\cdot, \cdot)$ in (5.5) we see that

$$p(\alpha|n, m) \propto \alpha^{a+n-1} e^{-b\alpha} \int_0^1 \eta^{\alpha-1} (1-\eta)^{m-1} d\eta.$$

Since this conditional distribution is difficult to sample from, Navarro et al. (2006) employ a *data augmentation*, where $p(\alpha|n, m)$ is viewed as a marginalization over

the joint density

$$p(\alpha, \eta|n, m) \propto \alpha^{a+n-1} e^{-b\alpha} \eta^{\alpha-1} (1-\eta)^{m-1}.$$

By using this joint distribution we deduce that

$$\alpha|\eta, n, m \sim Gamma(a+n-1, b-\log(\eta)) \tag{5.6}$$

and

$$\eta|\alpha, n, m \sim Beta(\alpha, m). \tag{5.7}$$

Since the DD is conjugate to the multinomial, it is straightforward to calculate the conditional posterior distribution over the $k$th cluster. Therefore the probability we require is $p(g_{jk} = 1|\underline{g}_{-j}, \alpha, \underline{X})$, the posterior probability that the $j$th object is assigned to the $k$th cluster, given the assignments for all other objects and a value for $\alpha$. Here $\underline{g}_{-j}$ denotes the cluster assignment for all other objects not including the $j$th. Then using Bayes rule we see that

$$\begin{aligned}
p(g_{jk} = 1|\underline{g}_{-j}, \alpha, \underline{X}) &\propto p(g_{jk} = 1|\underline{g}_{-j}, \alpha) p(\underline{X}_{j\cdot}|g_{jk} = 1, \underline{g}_{-j}, \underline{X}_{-j}) \\
&= p(g_{jk} = 1|\underline{g}_{-j}, \alpha) \int p(\underline{X}_{j\cdot}|\underline{\phi}_k) p(\underline{\phi}_k|\underline{g}_{-j}, g_{jk} = 1, \underline{X}_{-j}) d\underline{\phi}_k \\
&= p(g_{jk} = 1|\underline{g}_{-j}, \alpha) \frac{\int p(\underline{X}_{j\cdot}|\underline{\phi}_k) p(\underline{\phi}_k) \prod_{a' \in A_{-j}} p(\underline{X}_{a'}|\underline{\phi}_k) d\underline{\phi}_k}{\prod_{a' \in A_{-j}} p(\underline{X}_{a'\cdot})} \\
&= p(g_{jk} = 1|\underline{g}_{-j}, \alpha) \frac{\int p(\underline{\phi}_k) \prod_{a \in A} p(\underline{X}_{a\cdot}|\underline{\phi}_k) d\underline{\phi}_k}{\int p(\underline{\phi}_k) \prod_{a' \in A_{-j}} p(\underline{X}_{a'\cdot}|\underline{\phi}_k) d\underline{\phi}_k},
\end{aligned}$$

where $A = \{a : g_{ak} = 1\}$ and $A_{-j} = A - \{j\}$ is the set of objects in cluster $k$ including and not including the $j$th one respectively. Notice that with the marginal method we integrate out the $\underline{\phi}$ in (5.8). Since $p(g_{jk} = 1|\underline{g}_{-j}, \alpha)$ is the prior probability that a object $j$ from the DP belongs to cluster $k$, where $k$ may be an element of the currently observed clusters or a new cluster. It was shown by Neal (2000) that

$$p(g_{jk} = 1|\underline{g}_{-j}, \alpha) \propto \begin{cases} \frac{r_{-j,k}}{j+\alpha-1} & ; k \leq K_{-j} \\ \frac{\alpha}{j+\alpha-1} & ; \text{o.w,} \end{cases} \tag{5.8}$$

where $r_{-j,k}$ counts the number of objects (not including the $j$th) that are currently assigned to cluster $k$, and $K_{-j}$ denotes the number of clusters in the observed par-

tition over all objects except the $j$th. Next, expanding the integral

$$\frac{\int p(\underline{\phi}_k) \prod_{a \in A} p(\underline{X}_{a.}|\underline{\phi}_k) d\underline{\phi}_k}{\int p(\underline{\phi}_k) \prod_{a' \in A_{-j}} p(\underline{X}_{a'.}|\underline{\phi}_k) d\underline{\phi}_k},$$

we get

$$\begin{aligned} \int p(\underline{\phi}_k) \prod_{a \in A} p(\underline{X}_{a.}|\underline{\phi}_k) d\underline{\phi}_k &= \int \prod_{l=1}^{s} \phi_{kl}^{\beta + \sum_{j=1}^{m} X_{j.l} g_{jk} - 1} d\underline{\phi}_k \\ &= \frac{\prod_{l=1}^{s} \Gamma(\beta + \sum_{j=1}^{m} X_{j.l} g_{jk})}{\Gamma(s\beta + \sum_{j=1}^{m} \sum_{l=1}^{s} X_{j.l} g_{jk})}, \end{aligned}$$

and similarly

$$\begin{aligned} \int p(\underline{\phi}_k) \prod_{a' \in A_{-j}} p(\underline{X}_{a'.}|\underline{\phi}_k) d\underline{\phi}_k &= \int \prod_{l=1}^{s} \phi_{kl}^{\beta + \sum_{a' \in A_{-j}} X_{a'.l} g_{a'k} - 1} d\underline{\phi}_k \\ &= \frac{\prod_{l=1}^{s} \Gamma(\beta + \sum_{a' \in A_{-j}} X_{a'.l} g_{a'k})}{\Gamma(s\beta + \sum_{a' \in A_{-j}} \sum_{l=1}^{s} X_{a'.l} g_{a'k})}. \end{aligned}$$

Taken together we see that

$$\begin{aligned} \frac{\int p(\underline{\phi}_k) \prod_{a \in A} p(\underline{X}_{a.}|\underline{\phi}_k) d\underline{\phi}_k}{\int p(\underline{\phi}_k) \prod_{a' \in A_{-j}} p(\underline{X}_{a'.}|\underline{\phi}_k) d\underline{\phi}_k} &= \frac{\Gamma(s\beta + q_{-j,k}) \prod_{l=1}^{s} \Gamma(\beta + q_{.,k,l})}{\prod_{l=1}^{s} \Gamma(\beta + q_{-j,k,l}) \Gamma(s\beta + q_{.,k})} \\ &= w_{jk}, \end{aligned}$$

where $q_{-j,k,l} = \sum_{a' \in A_{-j}} X_{a'.l} g_{a'k}$ denotes the number of times that an object (not including the $j$th) currently assigned to cluster $k$ made response $l$, and $q_{-j,k} = \sum_{a' \in A_{-j}} \sum_{l=1}^{s} X_{a'.l} g_{a'k}$ denotes the total number of responses made by these objects. The terms $q_{.,k,l}$ and $q_{.,k}$ are defined similarly, except that the data for the $j$th object is not excluded. So, taking these results together with the $p(g_{jk} = 1|\underline{g}_{-j}, \alpha)$, we see that the conditional posterior for $g_{jk}$ is

$$p(g_{jk} = 1|\underline{g}_{-j}, \alpha, \underline{X}) \propto \begin{cases} w_{jk} \frac{r_{-j,k}}{j+\alpha-1} & ; k \leq K_{-j} \\ w'_{jk} \frac{\alpha}{(j+\alpha-1)} & ; \text{o.w,} \end{cases} \tag{5.9}$$

where $w'_{jk}$ is $w_{jk}$ with $q_{-j,k,l} = 0$. Next, to find the posterior for $\underline{\phi}_k$, we observe that

$$\begin{aligned} p(\underline{\phi}_k|\underline{g},\underline{X}) &\propto p(\underline{g},\underline{X}|\underline{\phi}_k)p(\underline{\phi}_k|\beta) \\ &\propto \left\{\prod_{j|g_{jk}=1} p(\underline{X}_{j.}|\underline{\phi}_k)\right\} p(\underline{\phi}_k|\beta) \\ &\propto \prod_{l=1}^{s} \phi_{kl}^{\sum_{j=1}^{m} X_{j.l} g_{jk}} \prod_{l=1}^{s} \phi_{kl}^{\beta-1} \\ &= \prod_{l=1}^{s} \phi_{kl}^{\sum_{j=1}^{m} X_{j.l} g_{jk} + \beta - 1}. \end{aligned}$$

Therefore

$$\underline{\phi}_k|\underline{g},\underline{X} \sim DD\left(\sum_{j=1}^{m} X_{j.l} g_{jk} + \beta\right). \tag{5.10}$$

Equations (5.6), (5.7), (5.9) and (5.10) define the Gibbs sampler. We call this the Dirichlet Process Multinomial Mixture using the Marginal method (DPMMM). Over time these samples converge to the full posterior of $(\alpha, \underline{g}, \underline{\phi})$. Again as with DPNMC/DPMMC, we can pick the most likely partition based on $p(\underline{g}|\alpha, \underline{\phi}, \beta, \underline{X})$ using a variation of the *integrated likelihood* ratio, see Section 4.3.2. We label this the Dirichlet Process Multinomial Mixture using the Marginal method for Clustering (DPMMMC).

We now provide a simulation to monitor the potential convergence times of the marginal and conditional methods under various values of $\alpha$. Under the conditional method we used a simpler version where we only have one profile since it is not easy to generalize the marginal method in this way. As before, convergence was assessed based on the block method criteria described in Section 4.3.4. We consider a data setup from the previous section: six brands (scenario 1 - three implanted clusters). Both methods were run in parallel with $\alpha$ set in the range [0.01,100]. From Table 5.2 we see that the conditional method had faster convergence times than the marginal. The difference in times was more marked for smaller values of $\alpha$. A possible reason is that the marginal method induces prior dependence between the $\underline{g}$ therefore increasing convergence times.

In addition to the computation time, we also monitored the deviance $D$ calculated as

$$D = -2\sum_{j=1}^{m} \log\left\{\sum_{k=1}^{m^*} \frac{r_k}{m} p(\underline{X}_{j.}|\underline{\phi}_k)\right\},$$

see Papaspiliopoulos and Roberts (2008) for further details. In a similar way to Papaspiliopoulos and Roberts (2008) we report the efficiency of both methods using the estimated integrated autocorrelation time, $\tau = 1 + 2\sum_{w=1}^{\infty} \rho_w$, where $\rho_w$ is the lag-$w$ autocorrelation of the monitored chain. Estimation of $\tau$ is a notoriously difficult problem as highlighted by Papaspiliopoulos and Roberts (2008). We use the suggestion by Papaspiliopoulos and Roberts (2008) where $\tau$ is estimated by summing estimated autocorrelations up to a fixed lag $L$, where $\tau << L << N$, and $N$ is the Monte Carlo sample size and was taken to be the size when the block method criterion was met[1], see Section 4.3.4. Approximate standard errors of the estimate can be obtained, see equation (3.19) in Sokal (1997). The results in Table 5.3 show that the difference in integrated autocorrelation times between the two methods is moderate, with greater variability observed for larger values of $\alpha$. However for larger values of $\alpha$ we could potentially sample directly from the parametric distribution $G_0$ instead of using the DP[2].

To specify an appropriate value of $\alpha$ at which point we can go parametric we make use of a variation of the maximum Kolmogorov distance. We compare the empirical distribution function (EDF) between the realization, $G$, from a DP and $G_0$ as follows:

1. Generate a realization $G$ from a the DP using Sethuraman representation, see Section 3.5. We now have a sequence of $\phi_1, \ldots, \phi_L$ and $w_1, \ldots, w_L$ to represent $G$

2. Order the $\phi_k$'s from smallest to largest, and using this ordering order the $w_k$'s

3. Compute $d_{(l)} = \sum_{k=1}^{l} w_{(k)} - G(\phi_{(k)})$, for $l = 1, \ldots, L$, where $G(\phi_{(k)}) = P[X < \phi_{(k)}]$ and $X \sim G$

4. Find $d_i = \max d_{(l)}$

5. Repeat 1-4 $N$ times and find the average distance $\bar{d} = \sum_{i=1}^{N} d_i/N$.

Figure 5.9 reports the $\bar{d}$ for various values of $\alpha$ based on $N = 100$, where $G_0 \equiv N(0, 1)$. Figure 5.9 shows that past $\alpha = 40$ we can directly sample from $G_0$ rather than use the DP, based on a $\bar{d} \leq 0.1$.

[1] We found $N$ between 500-1000 was sufficient here
[2] But at the cost of losing the clustering ability

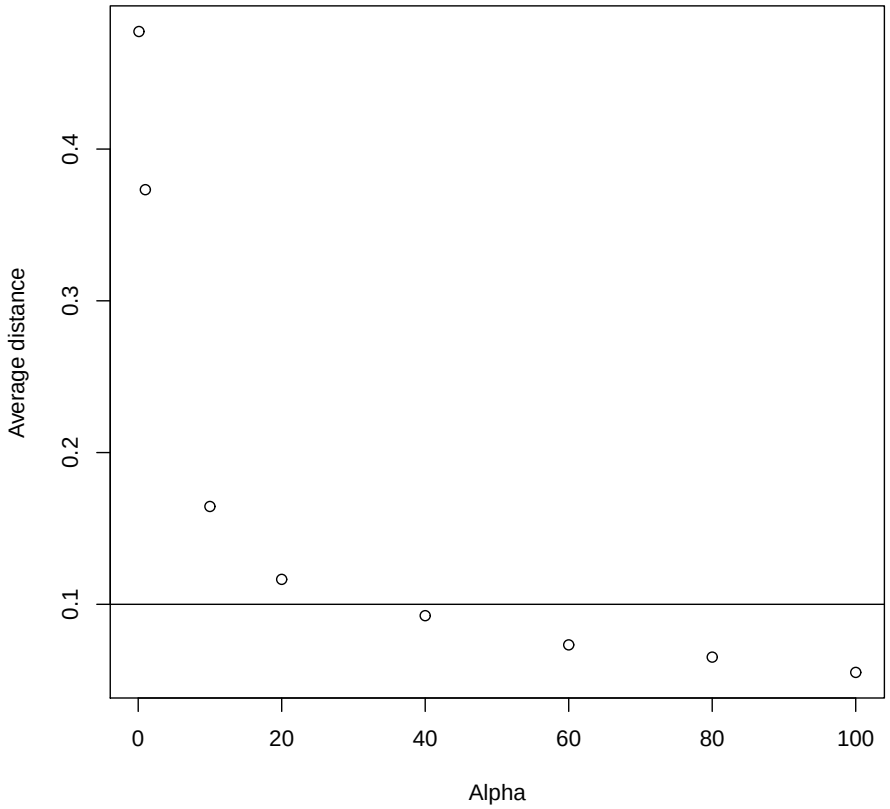


Figure 5.9: Estimated $\bar{d}$ for various values of $\alpha$ with $N = 100$.

| | Scenario 1 | | Scenario 2 | | Scenario 3 | |
|---|---|---|---|---|---|---|
| $\alpha$ | DPMMMC | DPMMC | DPMMMC | DPMMC | DPMMMC | DPMMC |
| 0.01 | 17.01 | 3.86 | 14.42 | 1.78 | 5.84 | 1.76 |
| 0.1 | 16.25 | 8.12 | 16.91 | 1.76 | 13.63 | 2.89 |
| 1 | 20.96 | 12.99 | 22.63 | 12.70 | 30.86 | 8.41 |
| 10 | 22.07 | 14.91 | 32.94 | 13.68 | 41.34 | 8.14 |
| 20 | 23.14 | 15.52 | 35.91 | 16.15 | 43.22 | 13.02 |
| 50 | 29.35 | 15.92 | 41.46 | 16.75 | 45.31 | 7.72 |
| 100 | 35.04 | 16.43 | 42.97 | 16.53 | 44.44 | 12.37 |

Table 5.2: Convergence times (secs) for DPMMMC and DPMMC. Simulation based on the six brands (scenario 1 - three clusters) dataset where $\hat{\beta} = 7$

| $\alpha$ | Scenario 1 | | Scenario 2 | | Scenario 3 | |
|---|---|---|---|---|---|---|
| | DPMMMC | DPMMC | DPMMMC | DPMMC | DPMMMC | DPMMC |
| 0.01 | 1.99(0.29) | 0.95(0.10) | 2.33(0.38) | 0.95(0.10) | 10.67(3.52) | 0.92(0.09) |
| 0.1 | 1.09(0.13) | 1.23(0.14) | 1.34(0.17) | 0.80(0.08) | 1.92(0.27) | 0.84(0.09) |
| 1 | 1.17(0.13) | 1.09(0.12) | 1.02(0.12) | 0.94(0.11) | 1.13(0.13) | 1.06(0.12) |
| 10 | 0.96(0.10) | 0.96(0.10) | 0.93(0.10) | 0.72(0.08) | 2.95(0.52) | 2.05(0.48) |
| 20 | 1.11(0.13) | 0.97(0.11) | 1.03(0.12) | 0.96(0.11) | 1.05(0.12) | 3.95(0.52) |
| 50 | 1.12(0.13) | 1.02(0.11) | 1.45(0.18) | 77.46(68.26) | 1.12(0.13) | 1.22(0.14) |
| 100 | 0.88(0.09) | 215.22(316.31) | 1.03(0.12) | 1.05(0.12) | 0.97(0.11) | 201.98(287.42) |

Table 5.3: Estimated integrated autocorrelated time for the deviance $D$. Estimated standard error in parentheses. Simulation based on the six brands (scenario 1 - three clusters) clusters dataset where $\hat{\beta} = 7$

## 5.6 Summary

From the simulation study in Section 5.4 we see that DPMMC offers some performance improvements over the other methods as well as DPNMC, particularly when we have a larger number of implanted clusters. Under a lower number of implanted clusters its performance is average in relation to the others. Since DPMMC models the data using its true distribution, we would expect superior performance in relation to DPNMC. One of the reservations with MNSC was that it outputs more erroneous clusters than needed, therefore often misleading to the researchers at Which? With DPMMC, we not only generate clusters from an infinite mixture model for adaptability, we also add extra information, e.g. the possible trials at Which?, through the profiles weights. The latter is a feature that is missing from the other methods we compared, and it seems to have been their downfall.

With regards to the two possible sampling mechanisms for the DPM, based on our simulations in the last section, we observe that the conditional method is more efficient than the marginal across the range of $\alpha$ values we explored.

Thus far focus on the specification of the $\alpha$ hyperparameters $(a, b)$ has been limited. Since $\alpha$ greatly influences the clustering behaviour we consider its properties in more detail in the next chapter. We also provide a framework for setting $(a, b)$ under both the informative as well as noninformative cases on the expected number of clusters.

# Chapter 6

# Learning the Clustering Structure

## 6.1 Introduction

In this chapter we consider the standard approaches that have been proposed in the literature for specifying a prior for the dispersion parameter $\alpha$. We then consider some theoretical properties followed by a proposed framework to capture the prior opinion on the expected number of clusters in an informative way using a percentile based method. In particular we focus on how we can adapt this framework in a number of ways that take account of both informative and noninformative setups. Under this adaptation we revisit the simulation study in Section 5.4 to observe any performance gains.

## 6.2 Current approaches

In both the DPNMC and DPMMC methods, see Chapters 4 and 5, we did not focus on the specification of the hyperparameters $(a, b)$ in the $Gamma(a, b)$ prior for $\alpha$. Instead, we set them to be small. This is an approach adopted by a number of authors, see Navarro et al. (2006). Placing a prior on $\alpha$ addresses the concerns in Antoniak (1974) regarding the DP model being rather restrictive if we set a value for $\alpha$ a priori. Other methods such as West et al. (1994) involve eliciting $(a, b)$ under strong prior knowledge for $\alpha$ or vary them over a wide range of $n$ values but place low probability on values of $n$ near one or $m$, where $n$ denotes a random variable for the number of district clusters and $1 \leq n \leq m$. One problem with this approach is that learning about $\alpha$ can be difficult, especially under a small sample size where the specification of $(a, b)$ will have a greater impact on the $\alpha$ posterior. Some other strategies are often based on approximations of the conditional mean and conditional

variance of $n$ given $\alpha$. However, when we know the prior mean and variance of $n$ we can use moment estimates of $(a, b)$ by equating the mean and variance to analytic approximations of their unconditional expectations of $\mathbb{E}[n]$ and $\mathbb{V}[n]$, see Jara et al. (2007).

The expected number of clusters sampled from a DP is given by

$$\mathbb{E}[n|\alpha, m] = \sum_{n=1}^{m} np(n|\alpha, m).$$

If we define $W_j$ as a random variable which equals one if we have a new cluster, and zero otherwise, then we see that $\mathbb{E}[W_j] = \alpha/(\alpha + j - 1)$. Therefore it follows that

$$\mathbb{E}[n|\alpha, m] = \alpha \sum_{j=1}^{m} \frac{1}{\alpha + j - 1}.$$

Using the fact that $\sum_{j=1}^{m} 1/j \sim \log(m)$ it follows that

$$\mathbb{E}[n|\alpha, m] \approx \alpha \log\left(\frac{m + \alpha}{\alpha}\right) \tag{6.1}$$

for large $m$ as noted by Antoniak (1974). We see from (6.1), as noted by Korwar and Hollander (1973), $n$ increases with $m$ in an approximately logarithmic fashion. In many applications $\alpha$ is unknown, so we either place a prior on $\alpha$, as we have done thus far, or estimate it based on the data using relationship (6.1). The latter approach is favoured by some authors, see Lijoi et al. (2007), and is often used when we have strong knowledge about $\mathbb{E}[n|\alpha, m]$. Here we can use (6.1) to find a suitable prior value for $\alpha$ by specifying our prior expectation, $\bar{n}$, of the number of clusters. Let $u = m/\alpha$, then (6.1) becomes

$$\bar{n} = \frac{m}{u} \log(u + 1). \tag{6.2}$$

We can solve equation (6.2) for $u$ using Newton's method as follows. Define

$$f(u) = \frac{1}{u} \log(u + 1) - \frac{\bar{n}}{m}.$$

Then

$$f'(u) = \frac{1}{u(u + 1)} - \frac{\log(u + 1)}{u^2}.$$

To find $u$, we iterate

$$u_{i+1} = u_i - \frac{f(u_i)}{f'(u_i)}, \quad i = 0, 1, \ldots$$

and then set $\alpha = m/u$.

Turning now to the Gamma prior on $\alpha$, one of the main problems here is in choosing appropriate values for $(a, b)$. Navarro et al. (2006) reasoned that the Gamma prior will be improper when $a = b = 0$, so they simply used the proper but diffuse prior $Gamma(10^{-10}, 10^{-10})$ instead. This would appear to be a suitable noninformative prior for $\alpha$ but, as we will see in the next section, this prior is problematic.

To help specify $(a, b)$ we first need to determine the problem context:

1. 'Noninformative', where we have limited a priori knowledge on the number of clusters expected in the data. In the context of the Which? product trials this could be a user trial with a mix of brands from the top, middle and bottom end of the market. Here the researcher may have limited knowledge on the possible cluster memberships present in the data.

2. 'Informative', where we have strong prior beliefs on the expected number of clusters in the data, e.g. an annually run brand trial at Which? where the researcher has some information on the expected number of clusters from past trials.

In the informative case we could either specify $\alpha$, by $a^*$ using relationship (6.1) and our belief on the expected number of clusters, or set $d$ in

$$\alpha | a, a^* \sim Gamma(d, d/a^*) \tag{6.3}$$

depending on the strength of our belief about $a^*$. In the latter, since we are placing a prior on $\alpha$, we allow for extra information from the data to update $\alpha$. In the noninformative case we could use (6.3), but with $d = 10^{-10}$, which is a similar setup to Navarro et al. (2006). Here, $\alpha$ will be centred around $a^*$ but with a huge variance thereby mimicking the usual noninformative case. However, when $\alpha$ is very small, or large, this could lead to an improper posterior. To see this consider

$$L(\alpha) \propto p(\underline{X}|\alpha) = \sum_g p(\underline{X}|g)p(g|\alpha)$$

and

$$p(\underline{g}|\alpha) = \int p(\underline{g}|\underline{u})p(\underline{u}|\alpha)d\underline{u}.$$

We see that

$$\begin{aligned} p(\underline{g}|\alpha) &= \int \prod_{k=1}^{m^*} \alpha(1-u_k)^{\alpha-1} \left[ u_k \prod_{z=1}^{k-1}(1-u_z) \right]^{r_k} d\underline{u} \\ &= \alpha^{m^*} \int \prod_{k=1}^{m^*} u_k^{(r_k+1)-1}(1-u_k)^{(\alpha+R_k)-1} d\underline{u} \\ &= \alpha^{m^*} \frac{\Gamma(r_1+1)\Gamma(\alpha+R_1)}{\Gamma(\alpha+R_0+1)} \frac{\Gamma(r_2+1)\Gamma(\alpha+R_2)}{\Gamma(\alpha+R_1+1)} \cdots \frac{\Gamma(r_{m^*}+1)\Gamma(\alpha)}{\Gamma(\alpha+r_{m^*}+1)}. \end{aligned} \tag{6.4}$$

Therefore

$$p(\underline{g}|\alpha) = \prod_{k=1}^{m^*} \gamma_k(\underline{g}, \alpha),$$

where

$$\begin{aligned} \gamma_k(\underline{g}, \alpha) &= \alpha \frac{\Gamma(r_k+1)\Gamma(\alpha+R_k)}{\Gamma(\alpha+R_{k-1}+1)} \\ &= \frac{\alpha r_k!}{\prod_{j=0}^{r_k}(\alpha+R_k+j)} = \frac{\alpha^{-r_k} r_k!}{\prod_{j=0}^{r_k}\left(1+\frac{R_k+j}{\alpha}\right)}. \end{aligned} \tag{6.5}$$

We now assess the behaviour of $p(\underline{g}|\alpha)$ as $\alpha \to 0$. If $k < m^*$ then $R_k > 0$, so from (6.5),

$$\frac{\gamma_k}{\alpha} \to \frac{r_k!}{\prod_{j=0}^{r_k}(R_k+j)} < \infty$$

so that $\gamma_k = \mathcal{O}(\alpha)$. Next, $k = m^*$ implies that $R_k = 0$, so

$$\gamma_k = \frac{r_k!}{\prod_{j=1}^{r_k}(\alpha+j)} \to 1.$$

Therefore

$$p(\underline{g}|\alpha) \to \begin{cases} 0 & ; \text{if } m^* > 1 \\ 1 & ; \text{if } m^* = 1, \end{cases}$$

as $\alpha \to 0$, from which it follows that

$$p(\underline{X}|\alpha) = \sum_{k=1}^{m^*} p(\underline{X}|\underline{g}_k)p(\underline{g}_k|\alpha) \to p(\underline{X}|null), \tag{6.6}$$

where $p(\underline{X}|null)$ is the likelihood under one cluster.

Furthermore, as $\alpha \to \infty$ $p(\underline{X}|\alpha)$ converges to the likelihood function in the model with no clustering, since $G \to G_0$ as $\alpha \to \infty$ almost surely. Thus $L(\alpha)$ tends to a positive limit at both zero and infinity. When $a = b = 0$ we see that $p(\alpha) \propto 1/\alpha$, therefore it follows that

$$p(\alpha|\underline{X}) \propto \begin{cases} p(\underline{X}|null)/\alpha & ; \text{for small } \alpha \\ p(\underline{X}|no\ clustering)/\alpha & ; \text{for large } \alpha. \end{cases} \tag{6.7}$$

From (6.7) we see that the posterior $p(\alpha|\underline{X})$ does not integrate to a finite limit when $\alpha \to 0$ or $\alpha \to \infty$, therefore leading to an improper posterior.

## 6.3 Alternative approaches

As we have seen from the previous section, in the noninformative setup, setting the hyperparameters $(a, b)$ very small causes problems in the $\alpha$ posterior. To address the near-impropriety of the $\alpha$ posterior we first observe that in any clustering situation we have $m$ objects to place amongst $n$ clusters. In the informative case, we elicit the probability of $n$ clusters from the experts in the domain of interest. Since experts find it difficult to quantify exact probabilities on the number of clusters, we propose to elicit only two pieces of information: the probability $p_{lower}$ of observing one cluster along with the probability $p_{upper}$ of observing more than $q_{upper} = \lceil c \log(m) \rceil$ clusters for some $c > 0$, where $\lceil x \rceil = \min\{h \in \mathbb{Z} | h \geq x\}$ is the ceiling function. Practically this makes sense, for example in the Which? product tests the researcher would have varied prior expectations for a larger, or smaller, number of clusters. So if they favour a larger number of clusters then $p_{upper}$ would be raised accordingly, and similarly $p_{lower}$ raised when a lower number of clusters is favoured. For practical purposes we set $c = 2$ to keep the upper bound threshold below $m$ for all $m \geq 2$ since we cannot observe more than $m$ clusters. For example, when we have six brands we elicit the probability of observing greater than or equal to four clusters to set $p_{upper}$, so the upper quantile is $q_{upper} = 4$. Since we know from (6.1) that the expected number of clusters from a DP increases in an approximately logarithmic fashion with $m$, there is some intuition behind the $q_{upper}$ cluster bound.

Having defined suitable quantiles, we can formally specify two nonlinear equations, $f_1(a,b)=0$ and $f_2(a,b)=0$, where

$$\begin{aligned} f_1(a,b) &= \int_0^\infty p(n=1|\alpha,m)p(\alpha)d\alpha - p_{lower} \\ &= \frac{b^a}{\Gamma(a)} z_{m1} \int_0^\infty \frac{\alpha^{a-1}e^{-b\alpha}}{\prod_{j=1}^{m-1}(\alpha+m-j)} d\alpha - p_{lower} \end{aligned}$$

and

$$\begin{aligned} f_2(a,b) &= \sum_{n=\lceil q_{upper} \rceil}^{m} \int_0^\infty p(n|\alpha,m)p(\alpha)d\alpha - p_{upper} \\ &= \frac{b^a}{\Gamma(a)} \sum_{n=\lceil q_{upper} \rceil}^{m} z_{mn} \int_0^\infty \frac{\alpha^{n+a-2}e^{-b\alpha}}{\prod_{j=1}^{m-1}(\alpha+m-j)} d\alpha - p_{upper}. \end{aligned}$$

Then these equations can be solved for $(a,b)$ by, for example, minimizing the objective function $f_3(a,b)=f_1^2(a,b)+f_2^2(a,b)$. Thus far we have shown how to solve these equations for $(a,b)$ when $p_{lower}$ and $p_{upper}$ are elicited from experts. However, in the noninformative setup, we can also work the other way i.e. solve for $p_{lower}$ and $p_{upper}$ when $(a,b)$ are given. This is particularly useful when tuning the DPMMC from Chapter 5 for improving classification performance. We refer to this setup as SCAL from herein.

For comparison purposes we use an alternative proposal for $(a,b)$ selection by Dorazio (2009). He assumed that the prior information about $n$ can be specified using $h(n)$. Dorazio (2009) assumed that in the absence of prior information the distribution of $n$ is discrete uniform $h(n)=1/m$, where $1 \leq n \leq m$. Under any $h(n)$ we can find a $Gamma(a,b)$ prior for $\alpha$ for which the induced prior for $n$

$$\begin{aligned} \pi(n|m,a,b) &= \int_0^\infty p(n|\alpha,m)p(\alpha)d\alpha \\ &= \frac{b^a}{\Gamma(a)} z_{mn} \int_0^\infty \frac{\alpha^{a-1}e^{-b\alpha}}{\prod_{j=1}^{m-1}(\alpha+m-j)} d\alpha \end{aligned}$$

closely matches $h(n)$. Using the Kullback-Leibler divergence between $h(n)$ and $\pi(n|m,a,b)$ gives

$$D_{KL} = \sum_{n=1}^{m} h(n) \log\left\{\frac{h(n)}{\pi(n|m,a,b)}\right\}.$$

Then computing values for $(a, b)$ that minimizes $D_{KL}$ yields a prior for $\alpha$ that matches our prior opinions expressed by $h(n)$. We call this method DORO. In the next section we compare the performance of SCAL and DORO.

## 6.4 Comparison of clustering methods

We reconsider the simulation study from Section 5.4, where the setup by Navarro et al. (2006) was used to set $(a, b)$. This is undesirable as we saw in the last section. Instead, in the six brand setup, we learn the 'best' combinations of $(a, b)$ through simulation under the first two performance measures[1] defined in Section 4.5[2]. We consider $a, b \in (0, 5]$. Some of the performance figures are shown in Table 6.1. After some careful exploration we selected $a = b = 1$ as, from Table 6.1, this configuration gives good all-round performance and corresponds to $p_{lower} = 0.34$ and $p_{upper} = 0.15$. We then treat the six brand case with $a = b = 1$ as the default. Under this configuration we solve the equations under $p_{lower} = 0.34$, $p_{upper} = 0.15$ to obtain candidate values for $(a, b)$ under $m$ brands, which we anticipate will give good all-round performance. Under this setting we obtain $a = 0.66$ and $b = 0.61$ for the ten brand case. Since the results for the six brand setup are scaled appropriately for any $m$, this provides an automated way of specifying $(a, b)$ for any $m$. Table 6.2 shows the two performance measures under the ten brand case with $(a, b)$ set using the SCAL and DORO as described in the last section. For comparison purposes we also add in other combinations of $(a, b)$ as in Table 6.1. From Table 6.2 we observe a gain in performance for a larger number of clusters with DORO whereas SCAL performs well under a medium, or smaller, number of clusters. We also see SCAL performs, on average, better across all other combinations of $(a, b)$. Inspecting the $(a, b)$ values in Table 1 of Dorazio (2009) we see that $b$ is much smaller than $a$ as $m$ grows. Therefore we expect the prior to favour a much larger number of clusters with increasing $m$ which could potentially lead to a data/prior clash, particularly when the number of clusters in the data is much lower than $m$. From (6.1) we know that the expected number of clusters sampled from a DP grows logarithmically in $m$ which is a more reasonable assumption.

Figure 6.1 presents the $(a, b)$ values under SCAL. We observe a stabilization of $(a, b)$ around $0.4 - 0.5$ with increasing $m$. Figure 6.1 also shows the fitted values for both the $(a, b)$ curves based on a negative exponential regression model. More

[1] Since comparisons are not made with K-means, the last performance measure, $p_3$, was dropped

[2] Alternatively, to account for Which? ideally seeking five classes of products we could set $p_{upper}$ a lot smaller for $q_{upper} \geq 5$

| | Scenario 1 | | Scenario 2 | | Scenario 3 | |
|---|---|---|---|---|---|---|
| $(a, b)$ | *Two Clusters* | *Three Clusters* | *Six Clusters* | *Two Clusters* | *Three Clusters* | *Six Clusters* | *Two Clusters* | *Three Clusters* | *Six Clusters* |
| $(0.025, 1)$ | (83)(8) | (79)(31) | (38)(58) | (67)(8) | (15)(19) | (0)(14) | (42)(2) | (15)(19) | (0)(14) |
| $(0.05, 1)$ | (84)(8) | (77)(36) | (42)(63) | (74)(12) | (19)(26) | (0)(13) | (44)(4) | (19)(26) | (0)(13) |
| $(1, 1)$ | (85)(31) | (68)(48) | (57)(70) | (68)(20) | (23)(30) | (2)(35) | (60)(10) | (23)(30) | (2)(35) |
| $(2, 1)$ | (79)(31) | (55)(45) | (62)(75) | (58)(22) | (17)(28) | (4)(41) | (51)(14) | (17)(28) | (4)(41) |
| $(4, 1)$ | (66)(31) | (43)(42) | (71)(78) | (37)(19) | (8)(28) | (25)(47) | (27)(14) | (8)(24) | (25)(47) |

Table 6.1: Performance figures under $m = 6$. Here $(\cdot)(\cdot)$ represents the % datasets with all clusters recovered, $p_1$, and the average number of correctly classified clusters in $(100 - p_1)\%$ clusters not completely recovered (i.e. when we fail to recover all clusters, we consider the % that were correctly classified amongst the recovered) respectively. We explore suitable values of $(a, b)$ where $\hat{\beta} = 7$

| | Scenario 1 | | | Scenario 2 | | | Scenario 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| *Method* | *Two Clusters* | *Five Clusters* | *Ten Clusters* | *Two Clusters* | *Five Clusters* | *Ten Clusters* | *Two Cluster* | *Five Clusters* | *Ten Clusters* |
| SCAL ($a = 0.66, b = 0.61$) | (91)(23) | (64)(62) | (14)(69) | (83)(28) | (14)(41) | (2)(53) | (55)(20) | (1)(23) | (0)(21) |
| DORO ($a = 0.53, b = 0.046$) | (85)(32) | (20)(36) | (15)(68) | (75)(24) | (2)(6) | (61)(66) | (46)(16) | (0)(3) | (10)(28) |
| ($a = 0.025, b = 1$) | (83)(4) | (64)(42) | (2) (54) | (68)(9) | (23)(39) | (0)(35) | (41)(10) | (0)(19) | (0)(9) |
| ($a = 0.05, b = 1$) | (87)(8) | (72)(54) | (1) (56) | (72)(9) | (28)(46) | (0)(39) | (50)(8) | (0)(27) | (0)(12) |
| ($a = 1, b = 1$) | (89)(33) | (71)(62) | (5) (64) | (78)(27) | (20)(50) | (0)(51) | (52)(19) | (2)(26) | (0)(23) |
| ($a = 2, b = 1$) | (83)(31) | (55)(63) | (6) (68) | (72)(23) | (19)(40) | (0)(56) | (51)(21) | (0)(27) | (0)(28) |
| ($a = 4, b = 1$) | (74)(29) | (37)(59) | (15)(74) | (65)(21) | (4)(36) | (2)(63) | (32)(21) | (0)(20) | (0)(37) |

Table 6.2: Performance figures under $m = 10$ for the SCAL, DORO method along with other (a,b) values for comparison purposes. Here $(\cdot)(\cdot)$ represents the % datasets with all clusters recovered, $p_1$, and the average number of correctly classified clusters in $(100 - p_1)\%$ clusters not completely recovered (i.e. when we fail to recover all clusters, we consider the % that were correctly classified amongst the recovered) respectively. We fix $\hat{\beta} = 7$ in both cases.

| | Scenario 1 | | | Scenario 2 | | | Scenario 3 | | |
|---|---|---|---|---|---|---|---|---|---|
| *Method* | *Two Clusters* | *Four Clusters* | *Eight Clusters* | *Two Clusters* | *Four Clusters* | *Eight Clusters* | *Two Cluster* | *Four Clusters* | *Eight Clusters* |
| SCAL ($a = 0.63, b = 0.61$) | (93)(31) | (68)(48) | (0)(34) | (97)(29) | (24)(36) | (0)(22) | (82)(21) | (1)(30) | (0)(13) |
| DORO ($a = 0.51, b = 0.027$) | (92)(27) | (50)(44) | (0)(10) | (85)(29) | (13)(27) | (0)(12) | (83)(18) | (1)(16) | (0)(11) |
| ($a = 0.025, b = 1$) | (78)(4) | (66)(37) | (0)(32) | (80)(10) | (34)(36) | (0)(19) | (59)(3) | (1)(22) | (0)(6) |
| ($a = 0.05, b = 1$) | (89)(9) | (67)(49) | (0)(35) | (89)(3) | (35)(39) | (0)(18) | (74)(1) | (6)(27) | (0)(6) |
| ($a = 1, b = 1$) | (95)(30) | (60)(52) | (0)(34) | (92)(25) | (36)(37) | (0)(26) | (84)(22) | (5)(23) | (0)(9) |
| ($a = 2, b = 1$) | (94)(33) | (57)(49) | (0)(34) | (93)(31) | (27)(35) | (0)(22) | (81)(23) | (5)(23) | (0)(9) |
| ($a = 4, b = 1$) | (93)(31) | (46)(47) | (0)(31) | (82)(26) | (15)(30) | (0)(22) | (67)(21) | (3)(21) | (0)(10) |

Table 6.3: Performance figures under $m = 16$ for the SCAL, DORO method along with other (a,b) values for comparison purposes. Here $(\cdot)(\cdot)$ represents the % datasets with all clusters recovered, $p_1$, and the average number of correctly classified clusters in $(100 - p_1)\%$ clusters not completely recovered (i.e. when we fail to recover all clusters, we consider the % that were correctly classified amongst the recovered) respectively. We fix $\hat{\beta} = 7$ in both cases.

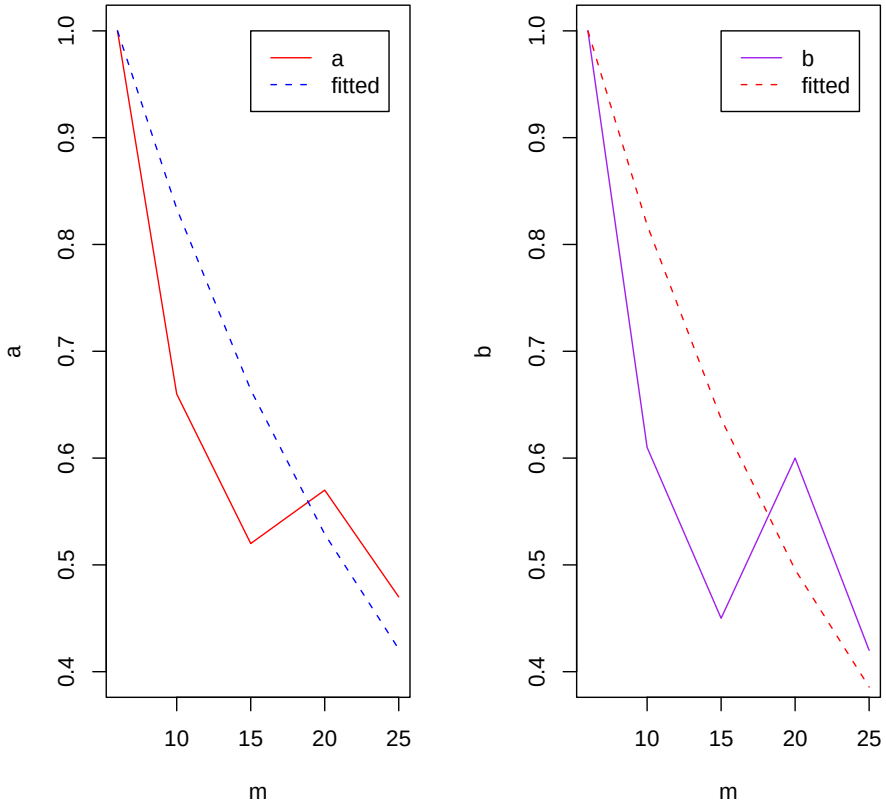


Figure 6.1: (left) Scaled $a$ (right) Scaled $b$ values under $m = 6$ with $p_{lower} = 0.34$ and $p_{upper} = 0.15$.

precisely $\hat{a} = e^{-0.046(m-6)}$ and $\hat{b} = e^{-0.050(m-6)}$. We can use these models to predict appropriate values of $(a, b)$ for $m \in [6, 25]$. To further demonstrate the effectiveness of this approach we also consider a 16 brand case. Using the approach described in Section 4.5 we simulate the 16 brand case with two, four and eight implanted clusters as follows:

Sixteen brands - two clusters

1. $(\underline{X}_{1,2,3,4,5,6,7,8})$ generated with $W_{16}C_1 = (\psi, \psi, \psi, 1, 1)$
2. $(\underline{X}_{9,10,11,12,13,14,15,16})$ generated with $W_{16}C_2 = (1, 1, \psi, \psi, \psi)$

Sixteen brands - four clusters

1. $(\underline{X}_{1,2,3,4})$ generated with $W_{16}C_1 = (\psi, \psi, 1, 1, 1)$
2. $(\underline{X}_{5,6,7,8})$ generated with $W_{16}C_2 = (1, \psi, \psi, 1, 1)$
3. $(\underline{X}_{9,10,11,12})$ generated with $W_{16}C_3 = (1, 1, \psi, \psi, 1)$
4. $(\underline{X}_{13,14,15,16})$ generated with $W_{16}C_4 = (1, 1, 1, \psi, \psi)$

Sixteen brands - eight clusters

1. $(\underline{X}_{1,2})$ generated with $W_{16}C_1 = (\psi, 1, 1, 1, 1)$
2. $(\underline{X}_{3,4})$ generated with $W_{16}C_2 = (\psi/2, \psi/2, 1, 1, 1)$
3. $(\underline{X}_{5,6})$ generated with $W_{16}C_3 = (\psi/3, \psi/3, \psi/3, 1, 1)$
4. $(\underline{X}_{7,8})$ generated with $W_{16}C_4 = (\psi, \psi/2, \psi/2, 1, 1)$
5. $(\underline{X}_{9,10})$ generated with $W_{16}C_5 = (1, 1, \psi, 1, 1)$
6. $(\underline{X}_{11,12})$ generated with $W_{16}C_6 = (1, 1, \psi/3, \psi/3, \psi/3)$
7. $(\underline{X}_{13,14})$ generated with $W_{16}C_7 = (1, 1, 1, \psi/2, \psi/2)$
8. $(\underline{X}_{15,16})$ generated with $W_{16}C_8 = (1, 1, 1, 1, \psi)$

As in Section 4.5 we took values of $\psi$ in the range $(10, 5, 3)$ for Scenarios 1-3 respectively. Using SCAL we find that $\hat{a} = 0.63$ and $\hat{b} = 0.61$ under the 16 brand case. Table 6.3 presents the two performance measures under this setting. From Table 6.3 we observe an improvement in performance using SCAL as opposed to DORO. However, the performance figures under $a = b = 1$ are similar to SCAL. It is interesting to observe that under eight implanted clusters, all configurations of

$(a, b)$ fail to recover any clusters. Since, from (6.1), the expected number of clusters grows logarithmically in $m$ provides an explanation for the poor performance in recovering a larger number of clusters.

We now reproduced the performance graphs from Section 5.4, but using SCAL to set $(a, b)$. We also add in DPNMC from Chapter 4 but with the same $(a, b)$ values we used in DPMMC to make comparisons fair. Figures 6.3-6.5 shows the performance measures for all methods under the six brands setup, and Figures 6.6-6.8 for ten brands. In addition, as before, we provide the posterior density for $\alpha$, see Figure 6.2, for the six brands (scenario 1 - three clusters) case along with the posterior means and standard deviations for the key parameters shown in Table 6.4. From Figure 6.2 it is clear that the posterior $\alpha$ values are more stable than before with a tighter SD owing partly to SCAL. Again, as in Section 5.4, notice the higher posterior weight placed on profiles 1, 3 and 4, which coincides with our simulated clusters, i.e. two from bottom, two from top and two from the middle market respectively. Unlike before the posterior SDs for these weights are a bit smaller indicating more certainty around these weights.

A number of interesting features can be observed from these figures. Firstly across most cases it is clear that using our framework shows additional performance gains over the standard DPNMC/DPMMC setups considered in Section 4.5 and 5.4. However, their performance is still average, under the six brands with the two implanted clusters scenario, as seen in Figure 6.3. As with the performance graphs in Sections 4.5 and 5.4 the performance of DPNMC/DPMMC is, at worst, average in relation to the other methods. As before, under performance measure two, DPMMC has better performance in relation DPNMC. In general we see that DPMMC performs better than DPNMC under more implanted clusters, particularly with ten brands. Notice in Figure 6.7 DPMMC has significant performance gains in relation to the other methods on measure one. This is particularly appealing for Which? since they seek five classes of products. Again, the drop in performance is more noticeable going from the second to the third scenario. Notice, in Figure 6.7, the large improvement in performance measure three with DPMMC in relation to the other methods. The same pattern is also observed in Figure 5.7 where $a = b = 10^{-2}$. Here the large improvement in performance, particularly with performance measure three, can in part be explained by the improper posterior for $\alpha$, see Section 6.2. Additionally, since we are unlikely to observe many outputted partitions with three clusters, particularly when $a = b = 10^{-2}$, performance measure three will naturally be more variable.

| | Prior | | Posterior | |
|---|---|---|---|---|
| *Parameter* | *Mean* | *SD* | *Mean* | *SD* |
| $\alpha$ | 1.00 | 1.00 | 1.38 | 0.81 |
| $\rho_1$ | 0.20 | 0.28 | 0.25 | 0.18 |
| $\rho_2$ | 0.20 | 0.28 | 0.07 | 0.18 |
| $\rho_3$ | 0.20 | 0.28 | 0.31 | 0.23 |
| $\rho_4$ | 0.20 | 0.28 | 0.30 | 0.20 |
| $\rho_5$ | 0.20 | 0.28 | 0.07 | 0.13 |

Table 6.4: Summary of the posterior mean and standard deviation for the key parameters in DPMMC for the six brands (three clusters) case.

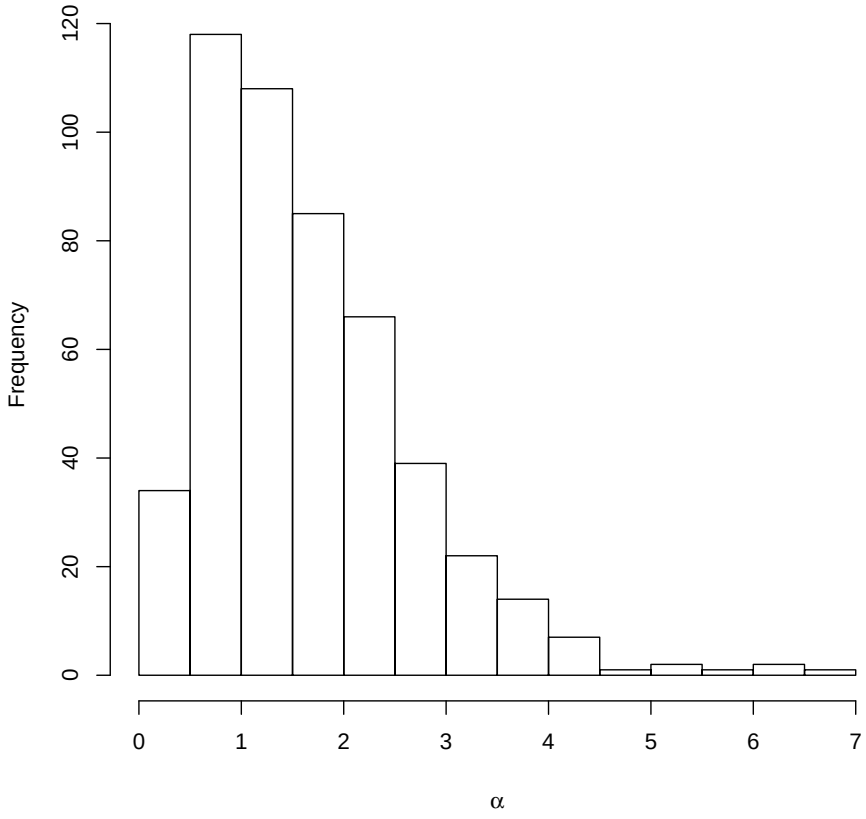


Figure 6.2: Posterior density for $\alpha$ under the six brands (scenario 1 - three clusters) case.

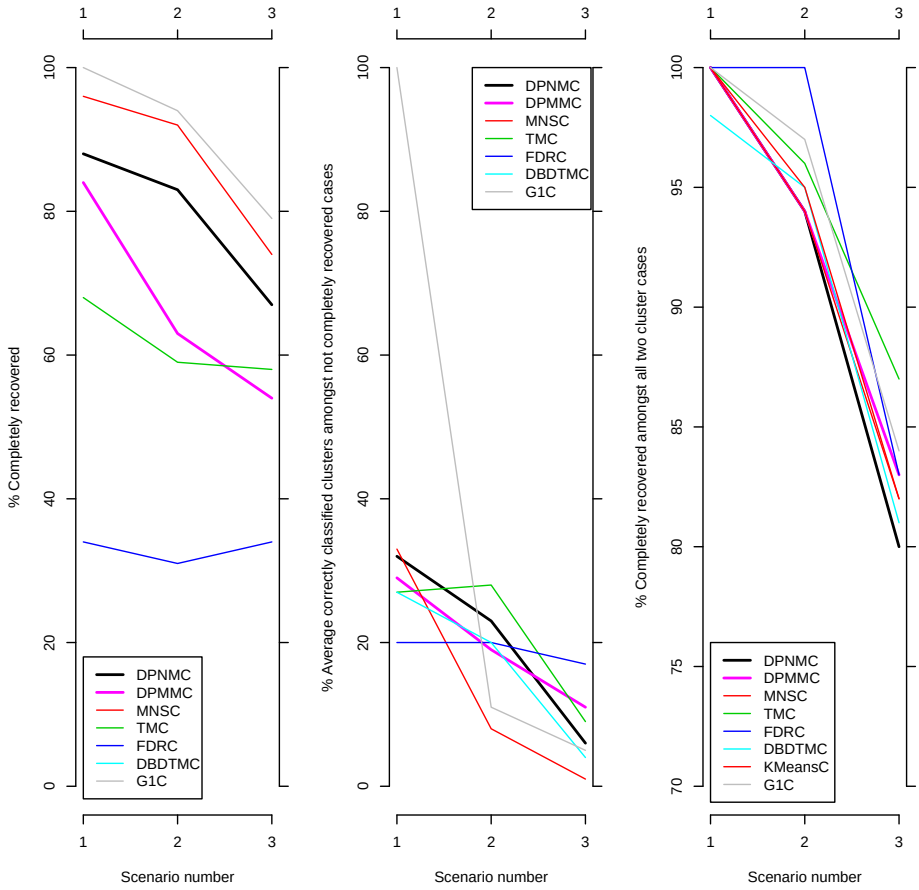


Figure 6.3: Performance of six brands (two implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 1$, $b = 1$ and $\hat{\beta} = 7$.

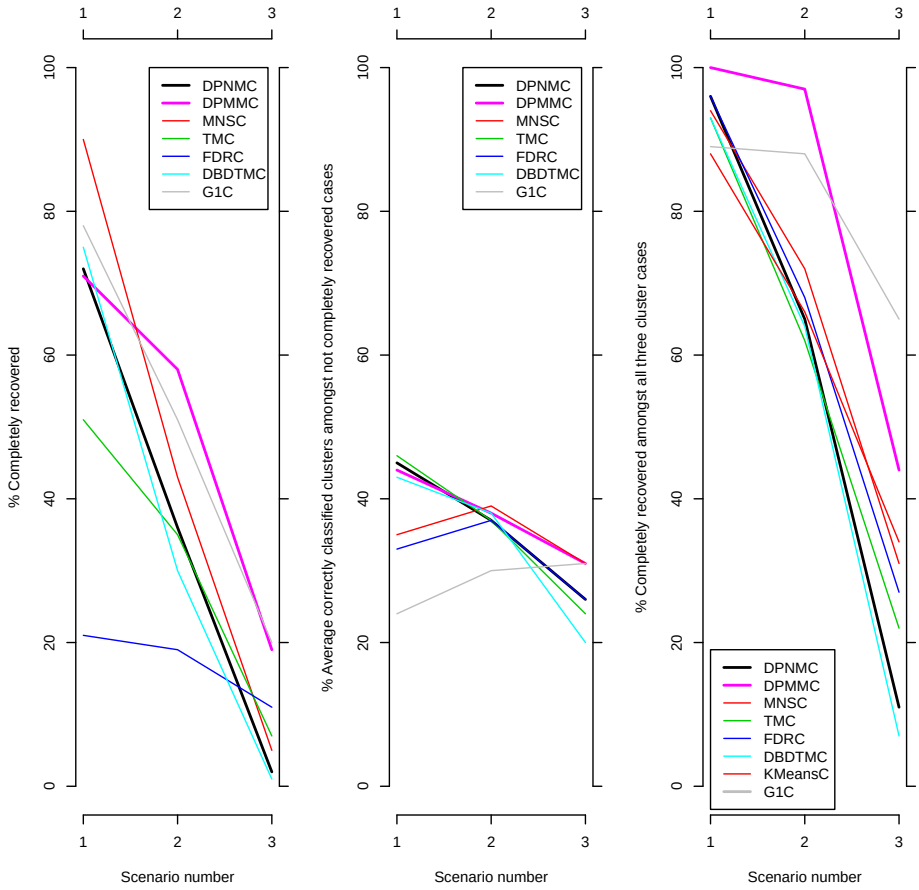


Figure 6.4: Performance of six brands (three implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 1$, $b = 1$ and $\hat{\beta} = 7$.

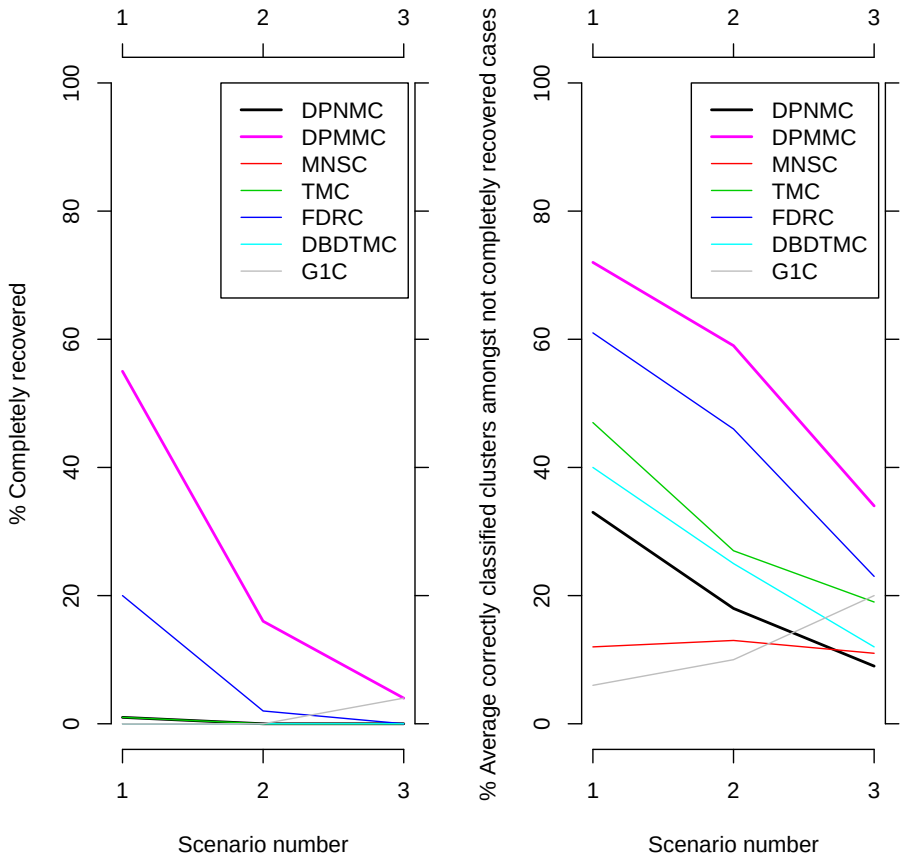


Figure 6.5: Performance of six brands (six implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 1$, $b = 1$ and $\hat{\beta} = 7$.

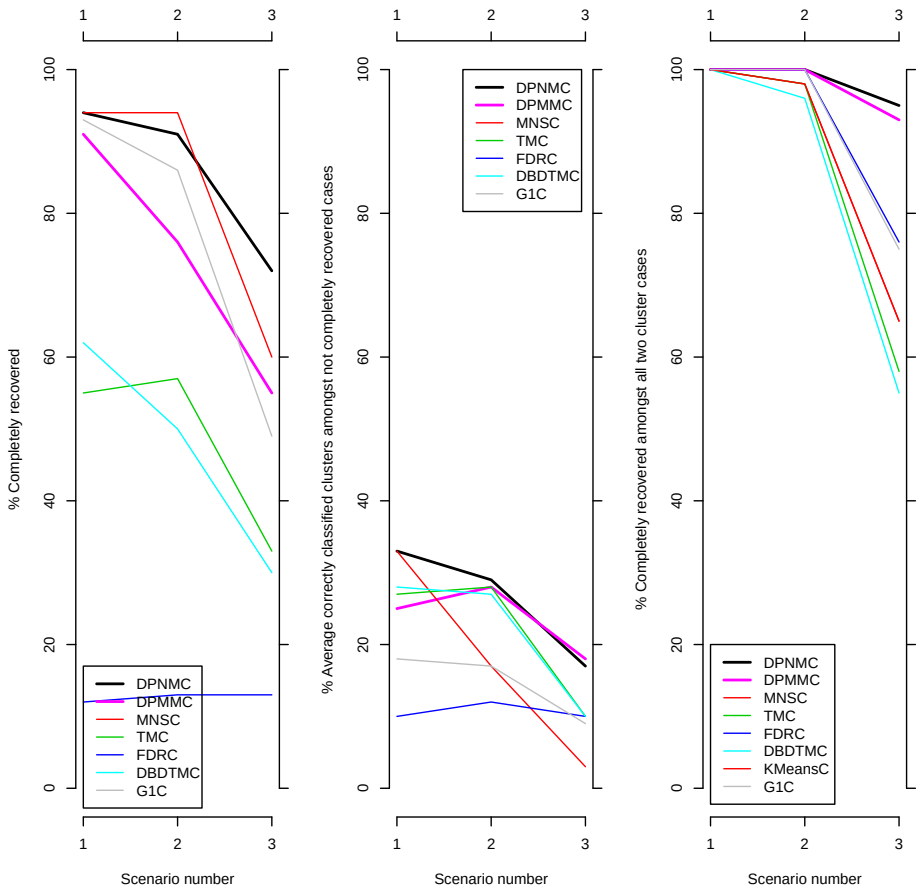


Figure 6.6: Performance of ten brands (two implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 0.66$, $b = 0.61$ and $\hat{\beta} = 7$.

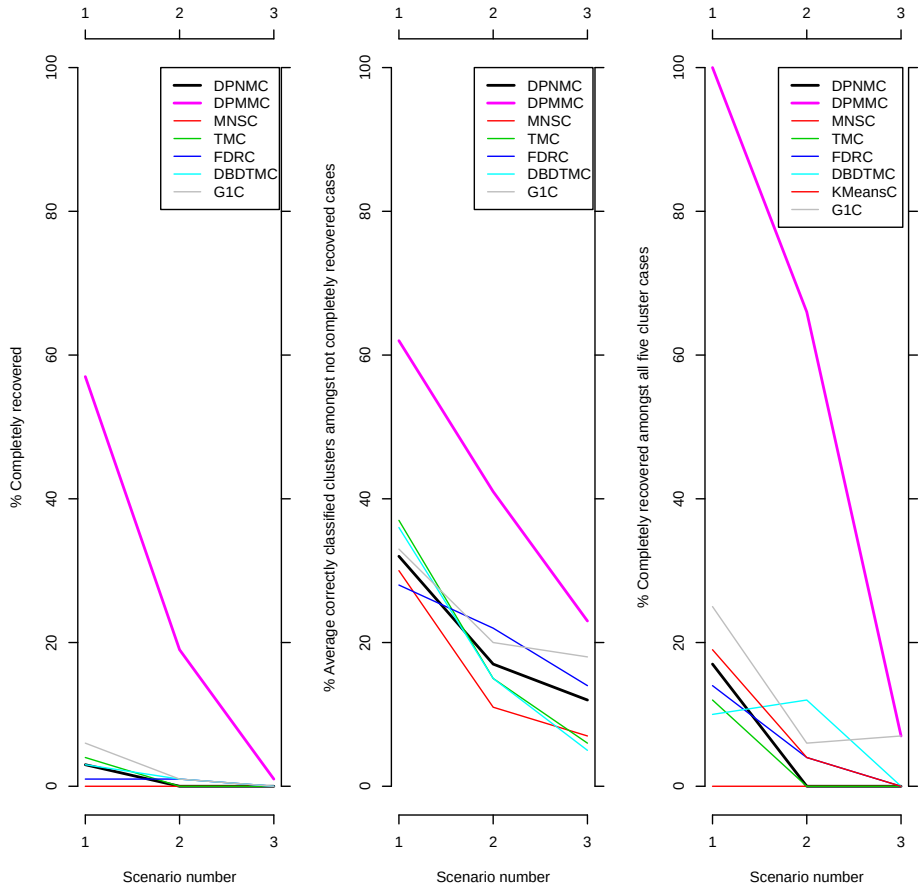


Figure 6.7: Performance of ten brands (five implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 0.66$, $b = 0.61$ and $\hat{\beta} = 7$.

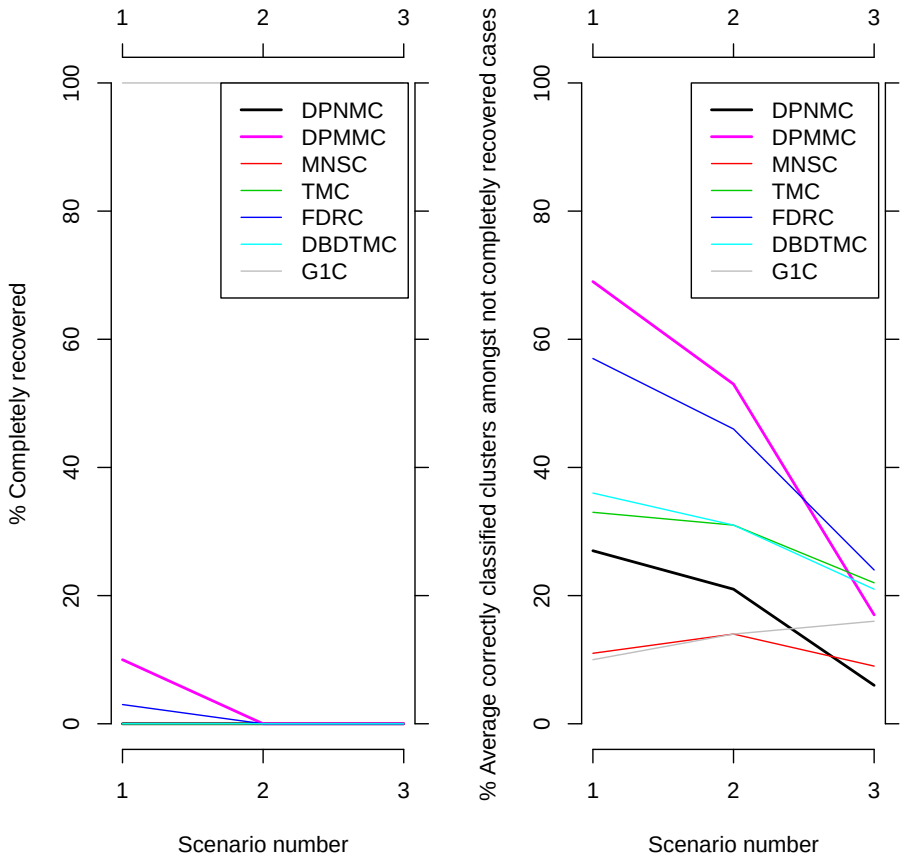


Figure 6.8: Performance of ten brands (ten implanted clusters) - The panel on the left indicates the results from the first, middle second and right third performance measure. Here, we took $a = 0.66$, $b = 0.61$ and $\hat{\beta} = 7$.

## 6.5 Summary

It is clear from the simulation study that $(a, b)$ specification using SCAL shows some performance gains, not only in relation to the other methods, but also in relation to the standard DPNMC/DPMMC setup introduced in Chapters 4-5. However, the performance of both DPNM/DPMMC is average under a lower number of implanted clusters. Although the performance of DPMMC across Figures 6.3-6.8 is average, it perform well under situations that will benefit Which?, particularly the setup in Figure 6.7 where good performance is observed on measure one. Considering DPMMC has improved performance for a larger number of implanted clusters, we believe a simulation with over ten brands will favour DPMMC more than the other methods. As we have seen from Chapter 1, one of the reservations with MNSC was that it was not stable in its final cluster solution, therefore often misleading researchers at Which?. By using DPNMC/DPMMC, we not only generate clusters from an infinite mixture model for improved adaptability and learning, but also incorporate extra prior information on observing a higher, or lower, number of clusters through SCAL. One of the main attractions of using SCAL is its automatic specification of $(a, b)$ for $m$ in a given range. This is an appealing feature for the Statisticians at Which? as it allows a robust way of clustering under the noninformative setup. It also allows researchers, under the informative setup, to specify their prior beliefs about the upper, or lower, number of clusters to estimate $(a, b)$. This is particularly useful for restricting the upper number of clusters to around five to fit in with Which?'s ideal number of blob classes.

# Chapter 7

# Conclusions and further work

## 7.1 Introduction

If the goal is to learn about complex variations amongst objects, e.g. how brands vary on an attribute question, then we require models that allow us to learn complex patterns of variation. To this end, the DPNMC and DPMMC provide a powerful method for representing the similarities and differences amongst objects on a particular attribute of interest. By adopting a DP prior, we are able to view observed clusters, not as a fixed structure, but rather as representatives of a latent arbitrarily rich structure. Additionally, by placing a prior over the dispersion parameter $\alpha$ we are able to learn about the cluster structure.

## 7.2 Contributions

We demonstrated the improvements one can expect by using the DPM for clustering over the other MCM based proposals. In particular we extended the standard DPM setup to account for the additional variation due to profiles in an experiment using DPMMC as illustrated in Chapter 5. This clearly gave some additional performance gains relative to the DPNMC as seen in Chapters 5-6. We also derived some theoretical properties related to the dispersion parameter $\alpha$ in Chapter 6 and provided a framework for selecting the hyperparameters $(a, b)$ in the Gamma prior for $\alpha$. The selection of these hyperparameters has received limited attention in the literature thus far. However, $\alpha$ is a crucial parameter since it determines the level of clustering and dispersion in the system, and careful setting of $(a, b)$ leads to improved performance, as seen in Chapter 6. Conventionally some authors set $a = b = 10^{-10}$ to signify the noninformative setup for $\alpha$, see Navarro et al. (2006). However, as

we have seen in Chapter 6, this leads to undesirable properties of the $\alpha$ posterior. Another aspect we considered was the MCMC computation for the DPM, which can be categorized into marginal and conditional methods. Both have their relative merits as we saw in Chapter 3 and Section 5.5. The DPNM/DPMM were based on an adapted variation of the Sethuraman's construction to make inferences possible, since it offers more flexibility when extending our base DPM model, see Chapter 3. Our variation allows us to to sample more efficiently from a DPM using the active and non-active components to address the ergodicity constraint, see Section 4.3.1. Finally, the application of DPMMC to our Which? problem in Section 1.1 shows promise in relation to their current MNSC method, see Sections 5.4 and 6.4.

## 7.3 Further work

The models presented in this thesis can be extended in several ways.

1. We briefly considered the GP as an alternative to the DP in Section 3.6. We could also investigate the clustering performance of other classes of nonparametric priors, such as the Pólya trees, see Kraft (1964), or Dirichlet diffusion trees, see Neal (2003).

2. We could extend the DPMM model so that, rather than having a set of $R$ profiles, we could have infinitely many profiles so that the distribution sampled from a DP is itself another DP. By doing so we allow for infinitely many types of profiles to be considered. This will create a double DP structure or a subset of the Hierarchical DP (HDP), see Teh et al. (2004).

3. One can look at various alternatives to the conditional method we used in constructing DPNM/DPMM, such as the Retrospective MCMC method proposed by Papaspiliopoulos and Roberts (2008) to address the problem of the imputation of an infinite-dimensional process using finite approximations. Papaspiliopoulos and Roberts (2008) demonstrate the retrospective sampling by simulating a realization $G$ from a DP. First we simulate $U_j \sim U[0,1]$, then set $g_{jk} = 1$ if and only if
$$\sum_{l=0}^{k-1} w_l < U_j \leq \sum_{l=1}^{k} w_l, \tag{7.1}$$
where $w_0 = 0$. Retrospective sampling simply exchanges the order of simulation between $U_j$ and pairs $(w_k, \phi_k)$. Rather than simulating $(\underline{w}, \underline{\phi})$ and then using $U_j$ in order to check condition (7.1), we instead simulate $U_j$ first then

pairs $(w_k, \phi_k)$. If given a $U_j$ we find that we need more $w_k$ to check condition (7.1), then we return to simulate pairs $(w_k, \phi_k)$ retrospectively until the condition is satisfied. The algorithm can be outlined as follows

(a) Simulate $w_1$ and $\phi_1$ and set $N = 1, j = 1$ and $w_0 = 0$.

(b) Repeat until $j > m$

i. Simulate $U_j \sim U[0, 1]$.

ii. If (7.1) is satisfied for some $k \leq N$, then set $g_{jk} = 1$, $\mu_j = \phi_k$, $j = j + 1$ and go to step (b)

iii. Else if (7.1) is not satisfied for any $k \leq N$, set $N = N + 1$, $k = N$ and simulate $w_k$ and $\phi_k$. Then go to step (ii).

We see that $N$ here keeps track of how far into the infinite sequence we have visited during the simulation.

We carry out a simulation study to contrast the performance of the retrospective with the standard conditional and marginal methods for sampling a realization $G$ from the DP as outlined in Chapter 3. We consider three configurations, namely 10,000, 5000 and 1000 samples from $G$, where $G_0 \sim N(0, 1)$. Under each configuration we let $\alpha = (10^{-3}, 10^{-2}, 0.1, 1, 5, 10, 20, 50, 80, 100)$. The sample generation times (secs) are shown in Figure 7.1. The results show that, under practical implementation, the relative time for the conditional method is significantly less.

4. With regard to the Which? problem, see Section 1.1, some trials at Which? involve a panel of five or so experts, each assessing the brands on various attributes. So rather than assuming a one-way ANOVA setup for the design we would need to consider a two-way ANOVA model where the experts are included as a factor in the model. One way to address this would be to use a multinomial logistic structure, so in the case of DPMM we would revise the model (5.2) by allowing the data $\underline{X}_{ji}|\underline{\theta}_{ji} \sim Mult(1, \underline{\theta}_{ji})$, where $\underline{\theta}_{ji} = (\theta_{ji1}, \ldots, \theta_{jis})$ denotes the probability with which the $j$th brand assessed by expert $i$ had the $l$th response. Then we allow the $\underline{\theta}_{ji}$ to have a multinomial logistic structure with experts and objects included as factors. It would be interesting to see if comparisons between DPMNC/DPMMC and the other clustering methods continue to hold under this more complex structure.

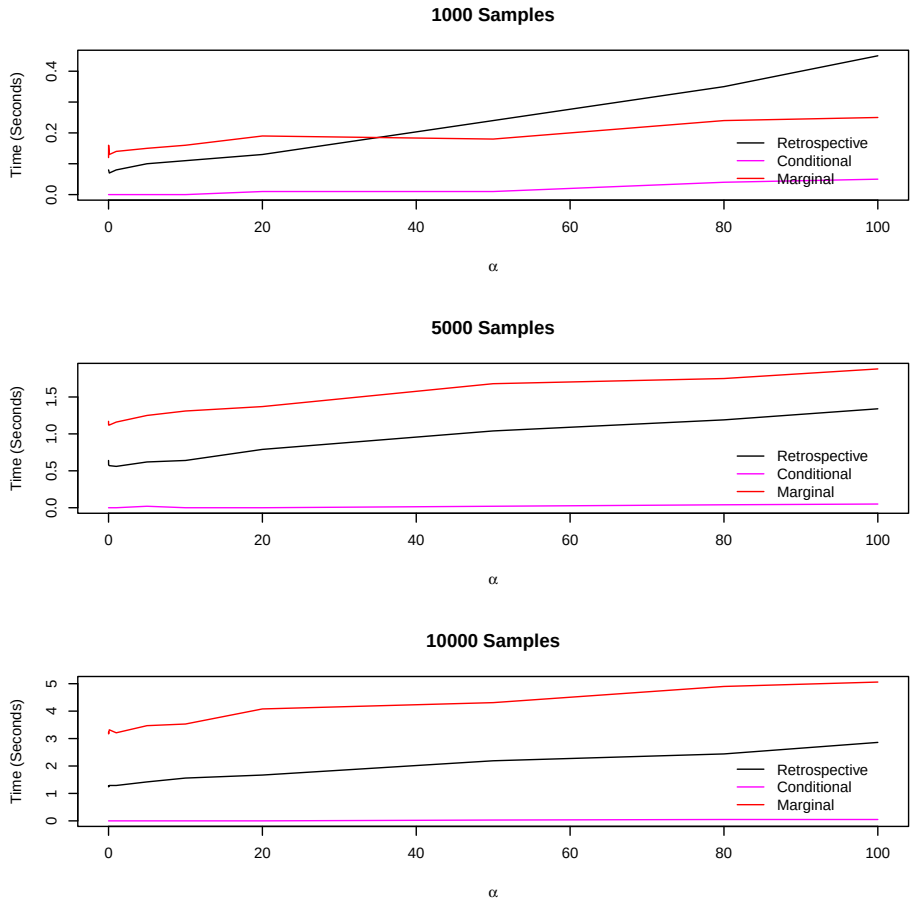


Figure 7.1: Sampling performance times (sec) for 10000, 5000 and 1000 samples based on a realization $G$ from a DP

## 7.4 Closing remarks

With regard to our original problem with the existing clustering methodology, MNSC, at Which?, see Section 1.1, the DPMMC offers an alternative and reliable statistical framework for capturing brand attribute differences. From the simulation studies in Sections 5.4 and 6.4 we saw the additional performance benefits of using DPMMC in relation to Which?'s existing MNSC methodology. The challenge now is in understanding how the new methodology DPMMC can be successfully implemented as a substitute for MNSC within the existing processes at Which?. To address this we plan to develop a commercialized version of DPMMC in Excel so that it can be used with a more user friendly interface.

We envisage that DPMs will gain even more popularity in coming years. A few possible extensions have already been mentioned in the previous sections, like the DPMM in Section 5.3. As we have seen, an important aspect of DPMs is in their implementation. Many approaches are driven by theoretical as well as computational concerns and will provide challenges for future research.

# Appendix A

# Appendix

## R-Function Help Files - DPM models

We provide details of the R help files for our DPNMC and DPMMC methods used in Chapters 4-6. We also include the function for estimating $\beta$ and a function that implements the framework for estimating the $\alpha$ hyperparameters $(a, b)$ as described in Chapter 6. The code was tested using R version 2.10.0 (Release 26-10-2009) and run on a Windows XP (SP2) platform.

### DPNMC

#### Description

This function performs the clustering of normal data based on Dirichlet Process Mixture Model for Clustering (DPNMC), see Chapter 4.

#### Usage

```
DPNMC=function(a=1,b=1,v0=0.001,sigmasq0=1,v1=0.001,sigmasq1=1,mu1star=1,
sigmasq2=1000,NumIterations=1000,Tol=0.001,dataIn)
```

## Arguments

| | |
|---|---|
| `a,b` | hyperparameters for the $\alpha$ posterior |
| `v0,sigmasq0` | hyperparameters for the $\sigma^2$ posterior |
| `mu1star,sigmasq2` | hyperparameters for the $\mu_0$ and $k_0$ posterior |
| `v1,sigmasq1` | hyperparameters for the $k_0$ posterior |
| `NumIterations` | specifies the total number of iterations of the Gibbs sampler with 20% discarded as the burn-in |
| `Tol` | specifies the tolerance for the missing probability mass such that $\sum_{h=1}^{L} w_h > 1 -$ `Tol` where $L$ is the number of samples required |
| `dataIn` | specifies the input data which should be entered in a matrix format with dimensions $m$ rows by $t$ columns, or $m$ objects with $t$ replicates |

## Details

See Chapter 4 for more details.

## Values

| | |
|---|---|
| `mean.alpha` | posterior mean for $\alpha$ based on the average of the after burn-in chain of $\alpha$ posterior samples |
| `mean.sigmasq` | posterior mean for $\sigma^2$ based on the average of the after burn-in chain of $\sigma^2$ posterior samples |
| `mean.k0` | posterior mean for $k_0$ based on the average of the after burn-in chain of $k_0$ posterior samples |
| `mean.mu0` | posterior mean for $\mu_0$ based on the average of the after burn-in chain of $\mu_0$ posterior samples |
| `partitionList` | list of posterior partitions (classification of objects into various clusters) ordered with the most frequently occurring first |
| `clusterMeanspartition` | list of cluster means, or centroids, for each of the occupied clusters |
| `clusterSTDEVpartition` | list of cluster standard deviations for each of the occupied clusters |
| `partitionListPCTOccurance` | vector of outputted posterior partition probabilities for `partitionList` outputted as a % with the most frequently occuring first |
| `partitionListHolderPCT` | list of posterior partition probabilities for each iteration after burn-in |
| `posteriorNullProbablity` | if a NULL partition exists (i.e all objects in the same cluster) the posterior NULL partition probability is outputted, otherwise NULL is returned |
| `posteriorNullPosition` | if a NULL partition exists (i.e all objects in the same cluster) the posterior NULL partition position in partitionList is outputted |

## Examples

```
dataSamples=NULL
##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance
for (l in (1:200)){

dataSamples[l]=(sample(c(rnorm(1,-4,1),rnorm(1,0,1),rnorm(1,8,1)),
1,replace=T))

}

##put samples in a matrix so that we have 10 object (rows) with 20
##replicates (columns)
dat.set=matrix(dataSamples,10,20)

##run DPNMC for 500 iterations, 100 burn-in, with a=b=1 and other
##parameters at their default values
DPNMC(a=1,b=1,NumIterations=500,dataIn=dat.set)
```

# DPMMC

## Description

This function performs clustering of multinomial data based on the Dirichlet Process Multinomial Mixture Model for Clustering (DPMMC), see Chapter 5.

## Usage

```
DPMMC=function(betaIn=1,a=1,b=1,NumIterations=1000,Tol=0.001,
dataIn,priorProfiles)
```

## Arguments

| | |
|---|---|
| `betaIn` | hyperparameters for $\underline{\phi}$ posterior |
| `a,b` | hyperparameters for $\alpha$ posterior |
| `NumIterations` | specifies the total number of iterations of the Gibbs sampler with 20% discarded as the burn-in |
| `Tol` | specifies the tolerance for the missing probability mass such that $\sum_{h=1}^{L} w_h > 1 -$ `Tol` where $L$ is the number of samples required |
| `dataIn` | specifies the input data which should be entered in a matrix format with dimensions $m$ rows by $t$ columns, or $m$ objects with $t$ replicates |
| `priorProfiles` | specifies the prior profiles which should be entered in a matrix format with dimensions $R$ profile rows by $s$ category columns |

## Details

See Chapter 5 for more details.

## Values

| | |
|---|---|
| `mean.alpha` | posterior mean for $\alpha$ based on the average of the after burn-in chain of $\alpha$ posterior samples |
| `mean.rho` | posterior mean for $\underline{\rho}$ based on the average of the after burn-in chain of $\underline{\rho}$ posterior samples |
| `partitionList` | list of posterior partitions (classification of objects into various clusters) ordered with the most frequently occurring first |
| `clusterMeanspartition` | list of cluster means, or centroids, for each of the occupied clusters |
| `clusterSTDEVpartition` | list of cluster standard deviations for each of the occupied clusters |
| `partitionListPCTOccurance` | vector of outputted posterior partition probabilities for `partitionList` outputted as a % with the most frequently occuring first |
| `partitionListHolderPCT` | list of posterior partition probabilities for each iteration after burn-in |
| `posteriorNullProbablity` | if a NULL partition exists (i.e all objects in the same cluster) the posterior NULL partition probability is outputted, otherwise NULL is returned |
| `posteriorNullPosition` | if a NULL partition exists (i.e all objects in the same cluster) the posterior NULL partition position in partitionList is outputted |

## Examples

```
##generate some multinomial data (Scenario 1 - 6 objects)
##with two implanted clusters and 20 counts per object
data.set=generateDataMult(m=6,t=20,categories=5,dataClusters=2,weights=c(1,10))

##set prior profile as in Section 5.4
priorProfile=matrix(,5,5)

##profile 1
priorProfile[1,]=c(0.3,0.3,0.13, 0.13, 0.13)
##profile 2
priorProfile[2,]=c(0.2,0.2,0.2, 0.2, 0.2)
##profile 3
priorProfile[3,]=c(0.13,0.13,0.13, 0.3, 0.3)
##profile 4
priorProfile[4,]=c(0.1,0.1,0.6, 0.1, 0.1)
##profile 5
priorProfile[5,]=c(0.3,0.13,0.13, 0.13, 0.3)

##run DPMMC for 500 iterations, 100 burn-in, other parameters at their
##default values
DPMMC(NumIterations=1000,Tol=0.001,dataIn=data.set,priorProfiles)
```

# pctN

## Description

This function gives the upper and lower percentile probabilities for the distribution of $n$

## Usage

```
pctN=function(m=6,a=1,b=1,c1=1,c2=2)
```

## Arguments

| | |
|---|---|
| `m` | specifies the number of objects |
| `a,b` | specifies the hyperparameters for the distribution of $\alpha$ |
| `c1, c2` | specifies constants in deriving the upper and lower quantiles |

## Details

See Chapter 6 for details

## Values

| | |
|---|---|
| `upper` | probability above the upperQuantile |
| `lower` | probability below the lowerQuantile |
| `upperQuantile` | the upper quantile value |
| `lowerQuantile` | the lower quantile value |

## Examples

```
##take 10 objects
m=10
##specify prior parameters for alpha based on the optimal simulation
##results for the six object case
a=1
b=1
##find the upper and lower probabilities based on these (a,b) values
PU=pctN(a,b)$upper
PL=pctN(a,b)$lower

##Here we fine PU=0.15 and PL=0.34

##construct two objective functions to minimize using PU=0.15
##and PL=0.34 as inputs, so that the appropriate (a,b) can be found for
##the 10 object case
objFunction=function(inp,PU=0.15,PL=0.34){

(pctN(exp(inp[1]),exp(inp[2]),m)$upper-PU)^2+
(pctN(exp(inp[1]),exp(inp[2]),m)$lower-PL)^2

}
##call the nlm function with initial starting values and specify 0 as the
##value of the objFunction at the minimum
exp(nlm(optimObjFunction,c(log(a),log(b)),typsize=c(0,0),fscale=0)$estimate)

##Using the output we find suitable estimates for (a,b)=(0.66, 0.61)
```

# getInitialBeta

## Description

This function gives the values from the integrated likelihood function for $\beta$

## Usage

```
getInitialBeta=function(beta,dataIn,weights=rep(1/5,5))
```

## Arguments

| | |
|---|---|
| `beta` | specifies the value for $\beta$ |
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `weight` | specifies prior weights for the $s$ catagories |

## Details

See Chapter 5 for details

## Values

| | |
|---|---|
| `fbeta` | negative value of the likelihood function evaluated at $\beta$ |

## Examples

```
##generate some multinomial data (Scenario 1 - 6 objects) with two implanted
##clusters and 20 counts per object
data.set=generateDataMult(m=6,t=20,categories=5,dataClusters=2,weights=c(1,10))

##minimize getInitialBeta function and find the MLE estimate for beta
output=nlm(getInitialBeta,1,hessian=TRUE)
beta=output$estimate
```

# R-Function Help Files - Other Clustering Methods

Here we provide the functions that implement the other clustering methods we adapted using the standard MCM procedures considered in Chapter 2

## MNSC

### Description

This function performs the method of normal scores clustering algorithm

### Usage

```
MNSC=function(dataIn,alpha=0.05,null=F)
```

### Arguments

| | |
|---|---|
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `alpha` | specifies the value of the $\alpha$ parameter |
| `null` | specifies a logical value. True if we are entering NULL data (all objects in the same cluster) or False otherwise |

### Details

See Chapter 1 for details

### Values

| | |
|---|---|
| `finalPartition` | final partition contains the allocation of the objects in their relevant clusters |
| `clusterMeans` | cluster means, or centroids, for the assigned clusters |

### Dependencies

No dependencies

## Examples

```
##put samples in a matrix 9 object (rows) with 20 replicate (columns)
dat.set=matrix(dataSamples,10,20)

##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance

##implant first cluster based on 20 replicates from a normal(-4,1)
for (i in (1:3)){
dat.set[i,]=rnorm(20,-4,1)
}

##implant second cluster based on 20 replicates from a normal(0,1)
for (i in (4:6)){
dat.set[i,]=rnorm(20,0,1)
}

##implant third cluster based on 20 replicates from a normal(8,1)
for (i in (7:9)){
dat.set[i,]=rnorm(20,8,1)
}

##run MNSC with other parameters at their default values
MNSC(dataIn=dat.set)
```

# TMC

## Description

This function performs the Tukey's method for clustering algorithm

## Usage

```
TMC=function(dataIn,alpha=0.05,null=F)
```

## Arguments

| | |
|---|---|
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `alpha` | specifies the value of the $\alpha$ parameter |
| `null` | specifies a logical value. True if we are entering NULL data (all objects in the same cluster) or False otherwise |

## Details

See Chapter 2 for details

## Values

| | |
|---|---|
| `finalPartition` | final partition contains the allocation of the objects in their relevant clusters |
| `clusterMeans` | cluster means, or centroids, for the assigned clusters |

## Dependencies

No dependencies

## Examples

```
##put samples in a matrix 9 object (rows) with 20 replicate (columns)
dat.set=matrix(dataSamples,10,20)

##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance

##implant first cluster based on 20 replicates from a normal(-4,1)
for (i in (1:3)){
dat.set[i,]=rnorm(20,-4,1)
}

##implant second cluster based on 20 replicates from a normal(0,1)
for (i in (4:6)){
dat.set[i,]=rnorm(20,0,1)
}

##implant third cluster based on 20 replicates from a normal(8,1)
for (i in (7:9)){
dat.set[i,]=rnorm(20,8,1)
}

##run TMC with other parameters at their default values
TMC(dataIn=dat.set)
```

# FDRC

## Description

This function performs the False discovery rate method for clustering algorithm

## Usage

```
FDRC=function(dataIn,delta=0.05,null=F)
```

## Arguments

| | |
|---|---|
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `delta` | specifies a value of the $\delta$ parameter |
| `null` | specifies a logical value. True if we are entering NULL data (all objects in the same cluster) or False otherwise |

## Details

See Chapter 2 for details

## Values

| | |
|---|---|
| `finalPartition` | final partition contains the allocation of the objects in their relevant clusters |
| `clusterMeans` | cluster means, or centroids, for the assigned clusters |

## Examples

```
##put samples in a matrix 9 object (rows) with 20 replicate (columns)
dat.set=matrix(dataSamples,10,20)

##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance

##implant first cluster based on 20 replicates from a normal(-4,1)
for (i in (1:3)){
dat.set[i,]=rnorm(20,-4,1)
}

##implant second cluster based on 20 replicates from a normal(0,1)
for (i in (4:6)){
dat.set[i,]=rnorm(20,0,1)
}

##implant third cluster based on 20 replicates from a normal(8,1)
for (i in (7:9)){
dat.set[i,]=rnorm(20,8,1)
}

##run FDRC with other parameters at their default values
FDRC(dataIn=dat.set)
```

# DBDTMC

## Description

This function performs the Duncan's Bayesian decision theoretic method for clustering algorithm

## Usage

```
DBDTMC=function(dataIn,k1=5,k2=1,iterations=500,null=F)
```

## Arguments

| | |
|---|---|
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `k1, k2` | specifies the loss due to a Type I (k1) and the loss due to a Type II error (k2) |
| `iterations` | specifies the total number of iterations from the posterior distribution |
| `null` | specifies a logical value. True if we are entering NULL data (all objects in the same cluster) or False otherwise |

## Details

See Chapter 2 for details

## Values

| | |
|---|---|
| `finalPartition` | final partition contains the allocation of the objects in their relevant clusters |
| `clusterMeans` | cluster means, or centroids, for the assigned clusters |

## Examples

```
##put samples in a matrix 9 object (rows) with 20 replicate (columns)
dat.set=matrix(dataSamples,10,20)

##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance

##implant first cluster based on 20 replicates from a normal(-4,1)
for (i in (1:3)){
dat.set[i,]=rnorm(20,-4,1)
}

##implant second cluster based on 20 replicates from a normal(0,1)
for (i in (4:6)){
dat.set[i,]=rnorm(20,0,1)
}

##implant third cluster based on 20 replicates from a normal(8,1)
for (i in (7:9)){
dat.set[i,]=rnorm(20,8,1)
}


##run DBDTMC with other parameters at their default values
DBDTMC(dataIn=dat.set)
```

# KMeansC

## Description

This function performs the K-means method for clustering

## Usage

```
KMeansC=function(dataIn,null=F,centers=3, nstart = 1)
```

## Arguments

| | |
|---|---|
| `dataIn` | specifies the data in matrix format $m$ objects (rows) by $t$ replicate (columns) |
| `centers` | specifies the number of clusters or a set of initial (distinct) cluster centres |
| `nstart` | specifies how many random sets should be chosen |
| `null` | specifies a logical value. True if we are entering NULL data (all objects in the same cluster) or False otherwise |

## Details

See Chapter 2 for details

## Values

| | |
|---|---|
| `finalPartition` | final partition contains the allocation of the objects in their relevant clusters |
| `clusterMeans` | cluster means, or centroids, for the assigned clusters |

## Examples

```
##put samples in a matrix 9 object (rows) with 20 replicate (columns)
dat.set=matrix(dataSamples,10,20)

##generate some normal data from a uniform mixture of three normals
##with means (-4,0,8) and unit variance

##implant first cluster based on 20 replicates from a normal(-4,1)
for (i in (1:3)){
dat.set[i,]=rnorm(20,-4,1)
}

##implant second cluster based on 20 replicates from a normal(0,1)
for (i in (4:6)){
dat.set[i,]=rnorm(20,0,1)
}

##implant third cluster based on 20 replicates from a normal(8,1)
for (i in (7:9)){
dat.set[i,]=rnorm(20,8,1)
}

##run KMeansC with other parameters at their default values
KMeansC(dataIn=dat.set)
```

# References


Allison, D. B., L., G. G., Heo, M., Fernandez, J. R., Lee, C. K., Prolla, T. A., and Weindruch, R. (2002). A mixture model approach for the analysis of microarray gene expression data. *Computational Statistics and Data Analysis*, 39:1–20. 15

Antoniak, C. (1974). Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *Ann. Statist*, 2:1152–1174. 29, 31, 87, 94, 95

Banfield, J. D. and Raftery, A. E. (1993). Model-based gaussian and non-gaussian clustering. *Biometrics*, 49:803–821. 25

Benjamini, Y. and Braun, H. (2002). John W. Tukey's contributions to multiple comparisons. *Ann. Statist*, 30:1576–1594. 19

Benjamini, Y. and Hochberg, T. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Statist. Soc., Series B*, 57:289–300. 11, 14, 15, 16

Berger, J. O. and Berry, D. A. (1988). Statistical analysis and the illusion of objectivity. *AME*, 76:156–165. 16

Berger, J. O. and Gugliemi, A. (2001). Bayesian testing of a parametric model versus nonparametric alternatives. *Ameri. Statist. Assoc*, 96:174–184. 29

Berger, J. O. and Wolpert, R. (1984). The likelihood principle. *Institute of Mathematical Statistics, Hayward, California*. 16

Bernardo, J. M. and Smith, A. F. (1994). *Bayesian Theory*. Wiley, New York. 29

Berry, D. A. (1988). Multiple comparisons, multiple tests, and data dredging: a Bayesian perspective. *Bayesian Statistics*, 3:79–94. 16, 17

Black, M. A. (2004). A note on the adaptive control of false discovery rates. *J. R. Statist. Soc., Series B*, 66:297–304. 16

Blackwell, D. and MacQueen, J. (1973). Ferguson distributions via Pólya urn schemes. *Ann. Statist*, 1:353–355. 35

Böckenholt, U. (2008). A latent class regression approach for the analysis of recurrent choices. *British Journal of Mathematical and Statistical Psychology*, 46:95–118. 26

Brix, A. (1999). Generalized gamma measures and shot-noise cox process. *Adv. Appl. Probab*, 31:929–953. 38

Celeux, G., Hurn, M., and Robert., C. P. (2000). Computational and inferential difficulties with mixture posterior distribution. *Ameri. Statist. Assoc*, 95:957–970. 25

Cox, D. (1972). Regression models and life-tables (with discussion). *J. R. Statist. Soc., Series B*, 34:187–220. 29

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *J. R. Statist. Soc., Series B*, 1:1–38. 25

Dorazio, M. (2009). On selecting a prior for the precision parameter of Dirichlet process mixture models. *J.Statist. Planning. Inf*, 10:10–16. 99, 100

Doss, D. and Huffer, F. (2003). Monte Carlo methods for Bayesian analysis of survival data using mixtures of Dirichlet process prior. *Journal of Computational and Graphical Statistics*, 12:282–307. 31

Duncan, D. B. (1965). A Bayesian approach to multiple testing. *Technometrics*, 7:171–222. 17

Dunnett, C. W. and Tamhane, A. C. (1992). A step-up multiple test procedure. *Ann. Statist*, 87:162–170. 11

Dunson, D. and Park, J. (2008). Kernel stick-breaking processes. *Biometrika*, 95:307–323. 38

Escobar, M. D. and West, M. (1994). Estimating normal means with the Dirichlet process prior. *Ameri. Statist. Assoc*, 89:268–277. 29

Escobar, M. D. and West, M. (1995). Bayesian density estimation and inference using mixtures. *Ameri. Statist. Assoc*, 90:577–588. 17, 47

Everitt., B. S. (1993). *Cluster analysis*. London: Edward Arnold. 26

Ferguson, T. S. (1973). A Bayesian analysis of some non-parametric problems. *Ann. Statist*, 1:209–230. 29, 31, 34

Fernando, R. L., Nettleton, D., Southey, B. R., Dekkers, J. C. M., Rothschild, M. F., and Soller, M. (2004). Controlling the proportion of false positives in multiple dependent tests. *Genetics*, 166:611–619. 15

Fisher, R. A. (1935). *The Design of Experiements.* Oliver and Boyd, Edinburg. 9

Frühwirth-Schnatter., S. (2001). Markov chain Monte carlo estimation of classical and dynamic switching mixture models. *Ameri. Statist. Assoc*, 96:194–209. 25

Ghosal, S., Ghosh, J., and Ramamoorthi, R. (1999). Posterior consistency of Dirichlet mixtures in density estimation. *Ann. Statist*, 27:143–158. 32

Gilbert, P. B. (2005). A modified false discovery rate multiple-comparisons procedure for discrete data, applied to human immunodeficiency virus genetics. *J. R. Statist. Soc., Series C*, 54:143–158. 12

Gilks, W., Richardson, S., and Spiegelhalter., D. (1995). *Markov chain Monte Carlo in Practice.* London: Chapman and Hall. 43, 53

Goodman, L. A. (1974). Exploratory latent structure analysis using both identifiable and unidentifiable models. *Biometrics*, 61:215–231. 26

Gordon, A. D. (1999). *Classification.* London: Chapman and Hall. 22, 24

Green, P. and Richardson, S. (1997). On Bayesian analysis of mixture models with an unknown number of components. *J. R. Statist. Soc., Series B.* 25

Griffin, J. and Steel, M. (2006). Order-based Dirichlet processes. *Ameri. Statist. Assoc*, 101:179–194. 31, 38

Hambleton, R. K., Swaminathan, H., and Rogers, H. J. (1991). *Fundamentals of Item Response Theory.* Newbury Park, CA: Sage Press. 26

Hartigan, J. (1975). *Clustering Algorithms.* Wiley, New York. 23

Hochberg, T. (1988). A sharper Bonferroni procedure for multiple tests of significance. *Biometrika*, 75:800–802. 11

Hochberg, Y. and Tamhane, C. (1987). *Multiple Comparison Procedures.* Wiley, New York. 8

Holm, S. (1979). A simple sequential rejective multiple test procedure. *Scand. J. Statistics*, 6:65–70. 11

Hsu, J. C. (1996). *Multiple Comparisons: Theory and Methods.* Chapman and Hall, London. 8

Ishwaran, H. and James, L. (2001). Gibbs sampling methods for stick-breaking priors. *Ameri. Statist. Assoc*, 96:161–173. 34, 35, 37, 38

Ishwaran, H. and James, L. (2002). Approximate Dirichlet process computing in finite normal mixtures: smoothing and prior information. *Journal of Computational and Graphical Statistics*, 11:508–532. 18, 47

Ishwaran, H. and Zarepour, M. (2000). Markov chain Monte Carlo in approximate Dirichlet and beta two-parameter process hierarchical models. *Biometrika*, 87:371–390. 35, 38

Jara, A., Garcia-Zattera, M., and Lesaffre, E. (2007). A Dirichlet process mixture model for the analysis of correlated binary response. *Computational Statistics and Data Analysis*, 51:5402–5415. 95

Jasra, A., Holmes, C. C., and Stephens, D. A. (2005). Markov chain Monte Carlo and the label switching problem in Bayesian mixture modelling. *Statistical Science*, 20(1):50–67. 25

Kalbfeisch, J. (1978). Non-parametric Bayesian analysis of survival time data. *J. R. Statist. Soc., Series B*, 40:214–221. 29

Kleinman, K. and Ibrahim, J. (1978). A semi-parametric Bayesian approach to generalized liner mixed models. *Statist. in Med*, 17:2579–2596. 29

Korn, E., Troendle, J., McShane, L., and Simon, R. (2004). Controlling the number of false discoveries: applications to high-dimensional genomic data. *J.Statist. Planning. Inf*, 124:379–398. 16

Korwar, R. and Hollander, M. (1973). Contributions to the theory of Dirichlet processes. *Ann. Prob.*, 1:705–711. 38, 95

Kraft, C. (1964). A class of distribution function processes which have derivatives. *J. Appl. Prob.*, 1:385–388. 116

Lavine, M. (1992). Some aspects of Pólya tree distributions for statistical modelling. *Ann. Statist*, 20:1222–1235. 29

Lavine, M. (1994). More aspects of Pólya tree distributions for statistical modelling. *Ann. Statist*, 22:1161–1176. 29

Lazarsfeld, P. F. and Henry, N. W. (1968). *Latent structure analysis.* Boston: Houghton Mill. 26

Lijoi, A., Mena, R., and Prunster, I. (2007). Controlling the reinforcement in Bayesian nonparametric mixture models. *J. R. Statist. Soc., Series B*, 69:715–740. 38, 39, 40, 54, 55, 69, 95

Lindley, D. and Smith, A. (1972). Bayes estimates for the linear model. *J. R. Statist. Soc., Series B*, 34:1–41. 30

Liu, J. S. (1996). Nonparametric hierarchical Bayes via sequential imputation. *Ameri. Sci.*, 24:911–930. 34

MacEachern, S. N. (2000). Dependent Dirichlet processes. *Technical report, Ohio State University, Department of Statistics.* 38

MacQueen, J. B. (1967). *Some Methods for Classification and Analysis of Multivariate Observations.*, volume 1. University of California Press. 23

Manly, K. F., Nettleton, D., and Hwang, J. T. G. (2004). Genomics, prior probability, and statistical tests of multiple hypotheses. *Genome Res*, 14:997–1001. 13

Morton, N. E. (1955). Sequential tests for the detection of linkage. *American Journal of Human Genetics*, 7:277–318. 13

Müller, P. and Quintana, F. (2004). Nonparametric Bayesian data analysis. *Statistical Science*, 19:95–110. 29

Navarro, D., Griffiths, T., Steyvers, M., and Lee, M. (2006). Modelling individual differences using Dirichlet processes. *Journal of Mathematical Psychology*, 50:101–122. 1, 60, 69, 78, 87, 94, 96, 100, 115

Neal, R. M. (2000). Markov chain sampling methods for Dirichlet process mixture models. *Journal of Computational and Graphical Statistics*, 9:249–265. 43, 88

Neal, R. M. (2003). Density modeling and clustering using Dirichlet diffusion trees. *Bayesian Statistics*, 7:619–629. 116

O'Brien, P. C. (1983). The appropriateness of analysis of variance and multiple-comparison procedure. *Biometrics*, 39:787–788. 26

O'Neill, R. T. and Wetherill, G. B. (1971). The present state of multiple comparisons methods (with discussion). *J. R. Statist. Soc., Series B*, 33:218–241. 3, 26

Owen, A. B. (2005). Variance of the number of false discoveries. *J. R. Statist. Soc., Series B*, 67:411–426. 16

Papaspiliopoulos, O. and Roberts, G. (2008). Retrospective Markov chain Monte Carlo methods for Dirichlet process hierarchical models. *Biometrika*, 95:169–186. 35, 38, 39, 91, 116

Papaspiliopoulos, O., Roberts, G., and Skold, M. (2007). A general framework for the parametrization of hierarchical models. *Statistical Science*, 22:59–73. 48

Pitman, J. and Yor, M. (1997). The two-parameter Poisson-Dirichlet distribution dervied from a stable subordinator. *Ann. Prob.*, 25:855–900. 38

Quintana, F. and Iglesias, P. (2003). Bayesian clustering and product partition models. *J. R. Statist. Soc., Series B*, 65:557–574. 30

Rothman, K. J. (1990). No adjustments are needed for multiple comparisons. *Epidemiology*, 1:43–46. 26

Schweder, T. and Spjtvoll, E. (1982). Plots of $p$-values to evaluate many tests simultaneously. *Biometrika*, 69:493–502. 11, 15

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Ann. Statist*, 4:639–650. 34

Shaffer, P. J. (1999). A semi-Bayesian study of Duncan's Bayesian multiple comparison procedure. *J.Statist. Planning. Inf*, 82:197–213. 17, 18, 19, 27

Simes, J. R. (1986). An improved Bonferroni procedure for multiple tests of significance. *Biometrika*, 73:750–754. 11

Sokal, A. (1997). Monte Carlo methods in statistical mechanics: Foundations and new algorithms. *In Functional Integration of NATO Adv. Sci. Inst. Ser. B Phys*, 361:131–192. 91

Stephens, M. (2000). Dealing with label switching in mixture models. *J. R. Statist. Soc., Series B*, 62:795–809. 25

Storey, J. D. (2002). A direct approach to false discovery rates. *J. R. Statist. Soc., Series B*, 64:479–498. 14, 15

Storey, J. D. (2003). The positive false discovery rate: a Bayesian interpretation and the $q$-value. *Ann. Statist*, 31:2013–2035. 15

Storey, J. D., Taylor, J. E., and Siegmund, D. (2004). Strong control conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *J. R. Statist. Soc., Series B*, 66:187–205. 16

Teh, Y., Jordan, M., Beal, M., and Blei, D. (2004). Hierarchical Dirichlet processes. *Technical report 653. Department of Statistics, University of California, Berkeley.* 116

Tocher, K. (1975). *The Art of Simulation.* Hodder and Stoughton. 52

Waller, R. A. and Duncan, D. B. (1969). A Bayes rule for the symmetric multiple comparison problem. *Ameri. Statist. Assoc*, 64:1484–1503. 17

Wedel, M., Bult, W. S., and Ramaswamy, V. (1999). A latent class Poisson regression model for heterogeneous count data with an application to direct mail. *Journal of Applied Econometrics*, 8:397411. 26

Welsch, R. E. (1977). Stepwise multiple comparisons procedures. *Ann. Statist*, 72:566–572. 11

Wenguang, S. and Tony, T. (2009). Large-scale multiple testing under dependence. *J. R. Statist. Soc., Series B*, 71:393–424. 16

West, M., Müller, P., and Escobar, M. (1994). *Hierarchical priors and mixture models, with applications in regression and density estimation.* John Wiley and Sons, New York. 37, 94

Westfall, P. H. and Johnson, W. O. (1997). A Bayesian perspective on the Bonferroni adjustment. *Biometrika*, 84:419–427. 17

Westfall, P. H. and Young, S. S. (1993). *Resampling-Based Multiple Testing.* John Wiley and Sons, New York. 8